Programming for Modular Reconfigurable Robots

Anna Gorbenko
Department of Mathematics and Mechanics
Ural State University
Ekaterinburg, Russia, 620083
Email: gorbenko.aa@gmail.com

Vladimir Popov
Department of Mathematics and Mechanics
Ural State University
Ekaterinburg, Russia, 620083
Email: Vladimir.Popov@usu.ru

***Abstract*—Composed of multiple modular robotic units, self-reconfigurable modular robots are metamorphic systems that can autonomously rearrange the modules and form different configurations for dynamic environments and tasks. Self-reconfiguration is to solve how to change connectivity among modules to transform the robot from the current configuration into the goal configuration within the restrictions of physical implementation. The existing reconfiguration algorithms used different methods, such as divide-and-conquer, graph matching etc, to reduce the reconfiguration cost. However, the optimal solution with least reconfiguration steps has never been reached. The optimal reconfiguration planning problem of finding the least number of reconfiguration steps to transform between two configurations is NP-complete. In this paper we describe an approach to solve this problem. This approach is based on constructing a logical models for considered problem.**

I. Introduction

Modular robotics has been the subject of much interest in the research community [1]. Using large numbers of simple modules to replace one complicated, special-purpose device provides benefits in terms of flexibility, robustness, and manufacturing cost. The challenge in these systems lies in controlling large numbers of low-powered, unreliable modules. Motion planning and shape formation for these systems is the main problem of such a difficult challenge.

Metamorphic robotic systems [2] can be viewed as a large swarm of connected robots which collectively act as a single entity. Potential applications of metamorphic systems composed of a large number of modules include:

- obstacle avoidance in highly constrained and unstructured environments;
- growing structures composed of modules to form bridges, buttresses, and other civil structures in times of emergency;
- envelopment of objects, such as recovering satellites from space;
- performing inspections in constrained environments such as nuclear reactors.

Self-reconfiguring robots were first proposed in [3]. In this planar system modules were heterogeneous and semi-autonomous. Other research focused on homogeneous systems with non-autonomous modules in two dimensions [4] – [7] and three dimensions [8] – [10]. In this type of system the modules are not capable of acting independently, and thus must remain connected. Five types of modular reconfigurable robotic systems have been proposed in the literature:

- robots in which modules are reconfigured using external intervention, e.g. [11] – [14];
- cellular robotic systems in which a heterogeneous collection of independent specialized modules are coordinated, e.g. [15] – [18];
- swarm intelligence in which there are generally no physical connections between modules, e.g. [19] – [22];
- modular robots composed of a few basic elements which can be composed into complex systems and used for various modes of locomotion, e.g. [23] – [25];
- fractal systems composed of modules with zero kinematic mobility, but which can walk over each other in discrete quanta due to changes in the polarity of magnetic fields, e.g. [5], [26].

In the present work, a metamorphic robotic system is a collection of independently controlled mechatronic modules, each of which has the ability to connect, disconnect, and climb over adjacent modules, e.g. [6]. A metamorphic system can dynamically reconfigure by the locomotion of modules over their neighbors. Thus they can be viewed as a collection of connected modular robots which act together to perform the given task. Composed of multiple modular robotic units, self-reconfigurable modular robots are metamorphic systems that can autonomously rearrange the modules and form different configurations for dynamic environments and tasks.

Modular reconfigurable robot programming can be substantially more challenging than normal robot programming due to:

- scale / number of modules;
- concurrency and asynchronicity, both in physical interactions and potentially at the software level;
- the local scope of information naturally available at each module.

For modular reconfigurable robots it is developed several specialized programming languages (e.g. [27], [28]). However, existing programming methods show relatively poor performance for reconfiguration planning problems. Note that reconfiguration planning problems play a central role for modular robots (e.g. [29] – [40]). Solutions for such problems lies at the heart of any control system of modular robots. Performance of such solutions is the base factor for the performance of the whole control system. So, the main challenges for modular robotic systems is an efficient planner. It has long been recognized that traditional methods are unsuitable due to the large search space and the blocking constraints imposed by realizable module design. To ease the planning problem, many groups have proposed different kinds of metamodules, groups of modules that act as a unit for planning or motion execution purposes, each specific to a particular module design [7], [38], [41], [42].

Poor performance for reconfiguration planning problems is not surprising, since such problems are computationally hard. In particular, in proved that the optimal reconfiguration planning problem of finding the least number of reconfiguration steps to transform between two configurations (ORP) is **NP**-complete. Therefore, we need some intelligent solution for this problem. However, the applying of distributed algorithms or any iterative procedure requires a great exchange of information between modules. This leads to the loss of solution accuracy and reduce performance. Therefore, it is desirable to solve ORP in a separate intelligent module which would generate a final solution represented by simple instructions. Note that the centralization of ORP solution allows to use some remote computing resources and makes the performance independent from computing resources of modules. When using such approach, programming of individual modules consists in

- processing of sensory information;
- transmission of sensory information;
- motor control;
- receiving instructions for actuators.

In this paper we describe an approach to solve ORP problem. This approach is based on constructing a logical models for considered problem.

II. Optimal Reconfiguration Planning Problem

Self-reconfiguration is to solve how to change connectivity among modules to transform the robot from the current configuration into the goal configuration within the restrictions of physical implementation. Depending on the hardware design, reconfiguration algorithms fall into two groups:

- reconfiguration for lattice-type modular robot and reconfiguration for chain-type modular robot. In lattice-type robot, modules lie in 2D or 3D grids;
- the reconfiguration is achieved through discrete movements of modules detaching from the current lattice location, moving along and surface of the robot and docking at the adjacent cells.

Example reconfiguration work includes [32] – [40] etc. In chain-type robots, modules can form moving chains and loops of any graph topology, and the reconfiguration is achieved through "connect" and "disconnect" operations between modules along with the joint motion of chains composed of several modules. Due to its difficulty, the chain-type reconfiguration has received less attention. Existing algorithms include [43] – [47] etc. The different geometric arrangement of modules between lattice-type and chain-type modular robots makes their reconfiguration planning mechanisms fundamentally different. The work in this paper is more focused on chain-type reconfiguration. For simplicity, we will use the term "modular robots" or simply "robots" to denote "the chain-type modular robots", and use "reconfiguration" to denote "chain-type reconfiguration" in the following.

The existing reconfiguration algorithms used different methods, such as divide-and-conquer [43], graph matching [44] etc, to reduce the reconfiguration cost. However, the optimal solution with least reconfiguration steps has never been reached. In [48] proved that the optimal reconfiguration planning problem of finding the least number of reconfiguration steps to transform between two configurations is **NP**-complete.

A. Configuration Representation

Before defining the optimal reconfiguration planning problem, we would describe representation of robot's configuration first. Two robots with the same graph topology can function differently if the modules are connected via different connectors (see e.g. [48]). To fully represent a robot's configuration, a special graph called C-Graph (Connector-Graph) is proposed in [48]. C-Graph is the extension of normal graph with differentiated connecting points. Each node has a finite number of ports that are internally labeled corresponding to the connectors of a module. A connection between module u's connector i and module v's connector j corresponds to an edge (i, j) between u and v.

In principle we could represent a robot's configuration as a C-Graph

$$G = (V, E),$$

$$V = \{v[1], v[2], \dots, v[n]\},$$

$$E = \{e[1], e[2], \dots, e[m]\},$$

where:

- each node $v[i] \in V$ represents the set

  $$v[i] = \{v[i,1], v[i,2], \dots, v[i,p_i]\}$$

  of connecting points of ith module, where p_i is the number of connecting points of ith module;
- each edge

  $$e[j] = (v[i_1,l_1], v[i_2,l_2]) \in E$$

  represents a connection between module i_1's connector l_1 and module i_2's connector l_2, where

  $$1 \le i_1 \le n, 1 \le i_2 \le n,$$

  $$1 \le l_1 \le p_{i_1}, 1 \le l_2 \le p_{i_2}.$$

B. Reconfiguration Actions

The two elementary reconfiguration actions are:

- making new connections;
- disconnecting current connections between modules for connectivity rearrangement.

The robot can bend its body through module joints, so any two modules with free connectors can potentially be aligned and dock with each other.

C. Optimal Reconfiguration Planning Problem

The reconfiguration planning problem is defined as how modules in one configuration rearrange into another using several sets of reconfiguration actions. Basically, what connections to make and what connections to disconnect so as to reconfigure from arbitrary one shape to another? Without loss of generality, we will always assume that the number of modules in the initial configuration is the same as that in the goal configuration.

During the reconfiguration process, the reconfiguration actions are most time- and energy-consuming, so it is a common practice to aim at minimizing the number of reconfiguration steps, i.e. the number of connect actions plus the number of disconnect actions. Therefore, the optimal reconfiguration planning problem is to find the least number of reconfiguration steps to transform from the initial configuration into the goal configuration.

Since the number of physical connections is predefined in the initial and goal configurations, the number of connect actions is fixed once the number of disconnect action is known, and vice versa. So we get that the optimal reconfiguration planning problem is to find the either one of the following metrics:

- least number of connect actions;
- least number of disconnect actions;
- least number of reconfiguration steps (i.e., the number of connect actions plus the number of disconnect actions).

For given two connected C-Graphs

$$I = (V, E_1)$$

and

$$G = (V, E_2)$$

we say that there exists a reconfiguration plan with at most k reconfiguration steps if and only if there exists a sequence of $r \leq k$ connect and disconnect actions such that starting from I we obtain G and applying each of this connect and disconnect actions we obtain a connected C-Graph. The decision version of optimal reconfiguration planning problem is formulated as the following problem.

OPTIMAL RECONFIGURATION PLANNING PROBLEM (ORP):

INSTANCE: *C-Graphs $I = (V, E_1)$ and $G = (V, E_2)$, a given integer k.*

QUESTION: *Whether there exists a reconfiguration plan for C-Graphs I and G with at most k reconfiguration steps?*

III. LOGICAL MODEL OF ORP

The propositional satisfiability problem (PSAT) is a core problem in mathematical logic and computing theory. Propositional satisfiability is the problem of determining if the variables of a given boolean function can be assigned in such a way as to make the formula evaluate to true. PSAT was the first known **NP**-complete problem, as proved by Stephen Cook in 1971 [49]. Until that time, the concept of an **NP**-complete problem did not even exist. Considered also different variants of the satisfiability problem. For instance, Satisfiability (SAT) is the problem of determining if the variables of a given boolean function in conjunctive normal form can be assigned in such a way as to make the formula evaluate to true. In practice, the satisfiability problem is fundamental in solving many problems in automated reasoning, computer-aided design, computer-aided manufacturing, machine vision, database, robotics, integrated circuit design, computer architecture design, and computer network design. Traditional methods treat the satisfiability problem as a discrete, constrained decision problem.

A. Reduction to PSAT

Consider a set of C-Graphs

$$\{G[q] = (V, E[q] \mid 0 \leq q \leq k\},$$

where

$$E[q] = \{e[q,1], e[q,2], \ldots, e[q, m_q]\},$$

each edge

$$e[q,j] = (v[i_1, l_1], v[i_2, l_2]) \in E[q]$$

represents a connection between module i_1's connector l_1 and module i_2's connector l_2, where

$$1 \leq i_1 \leq n, 1 \leq i_2 \leq n, 1 \leq l_1 \leq p_{i_1}, 1 \leq l_2 \leq p_{i_2}.$$

Let $G[0] = I$, $G[k] = G$. Now consider a set of boolean variables

$$\{x[q, i_1, i_2, i_3, i_4] \mid 0 \leq q \leq k, 1 \leq i_1 \leq n,$$

$$1 \leq i_2 \leq p_{i_1}, 1 \leq i_3 \leq n, 1 \leq i_4 \leq p_{i_3}\}.$$

Suppose that relation

$$x[q, i_1, i_2, i_3, i_4] = 1$$

means that

$$(v[i_1, i_2], v[i_3, i_4]) \in E[q].$$

Consider following boolean function:

$$\psi[q] \rightleftharpoons \bigvee_{\substack{1 \le s_1 \le n,\\ 1 \le s_2 \le p_{i_1},\\ 1 \le s_3 \le n,\\ 1 \le s_4 \le p_{i_3}}} \bigwedge_{\substack{1 \le i_1 \le n,\\ 1 \le i_2 \le p_{i_1},\\ 1 \le i_3 \le n,\\ 1 \le i_4 \le p_{i_3},\\ i_1 \ne s_1,\\ i_2 \ne s_2,\\ i_3 \ne s_3,\\ i_4 \ne s_4,\\ i_1 \ne s_3,\\ i_2 \ne s_4,\\ i_3 \ne s_1,\\ i_4 \ne s_2}} x[q, i_1, i_2, i_3, i_4] =$$

$$x[q+1, i_1, i_2, i_3, i_4].$$

It is easy to see that boolean function $\psi[q]$ is satisfiable if and only if $G[q] = G[q+1]$ or C-Graph $G[q+1]$ obtained from $G[q]$ by one connect or disconnect action. Therefore, it is easy to see that boolean function

$$(\bigwedge_{(v[i_1,i_2], v[i_3,i_4]) \in E[0]} x[0, i_1, i_2, i_3, i_4] = 1)\wedge$$

$$(\bigwedge_{(v[i_1,i_2], v[i_3,i_4]) \notin E[0]} x[0, i_1, i_2, i_3, i_4] = 0)\wedge$$

$$(\bigwedge_{(v[i_1,i_2], v[i_3,i_4]) \in E[k]} z[k, i_1, i_2, i_3, i_4] = 1)\wedge$$

$$(\bigwedge_{(v[i_1,i_2], v[i_3,i_4]) \notin E[k]} z[k, i_1, i_2, i_3, i_4] = 0)\wedge$$

$$(\bigwedge_{\substack{1 \le i_1 \le n,\\ 1 \le i_2 \le p_{i_1},\\ 1 \le i_3 \le n,\\ 1 \le i_4 \le p_{i_3},}} (\bigvee_{j=1}^{n^2} w[j, i_1, i_2, i_3, i_4]))\wedge$$

$$(\bigwedge_{\substack{1 \le i_1 \le n,\\ 1 \le i_2 \le p_{i_1},\\ 1 \le i_3 \le n,\\ 1 \le i_4 \le p_{i_3},}} (\bigwedge_{\substack{1 \le j_1 \le n^{n^2},\\ 1 \le j_2 \le n^{n^2},\\ j_1 \ne j_2}} (\neg w[j_1, i_1, i_2, i_3, i_4] \vee$$

$$\neg w[j_2, i_1, i_2, i_3, i_4])))\wedge$$

$$(\bigwedge_{\substack{1 \le i_1 \le n,\\ 1 \le i_2 \le p_{i_1},\\ 1 \le i_3 \le n,\\ 1 \le i_4 \le p_{i_3},\\ 1 \le i_5 \le n,\\ 1 \le i_6 \le p_{i_1},\\ 1 \le i_7 \le n,\\ 1 \le i_8 \le p_{i_3},\\ (i_1,i_3) \ne (i_5,i_7),\\ (i_1,i_3) \ne (i_7,i_5)}} \bigwedge_{j=1}^{n^2} (\neg w[j, i_1, i_2, i_3, i_4] \vee$$

$$\neg w[j, i_5, i_6, i_7, i_8]))\wedge$$

$$(\bigwedge_{\substack{1 \le i_1 \le n,\\ 1 \le i_2 \le p_{i_1},\\ 1 \le i_3 \le n,\\ 1 \le i_4 \le p_{i_3}}} \bigwedge_{j=1}^{n^2} w[j, i_1, i_2, i_3, i_4] =$$

$$w[j, i_3, i_2, i_1, i_4])\wedge$$

$$(\bigwedge_{\substack{1 \le i_1 \le n,\\ 1 \le i_2 \le p_{i_1},\\ 1 \le i_3 \le n,\\ 1 \le i_4 \le p_{i_3},\\ 1 \le i_5 \le n,\\ 1 \le i_6 \le n}} w[(i_5 - 1)n + i_6, i_1, i_2, i_3, i_4] \rightarrow$$

$$x[k, i_1, i_2, i_3, i_4] = z[k, i_5, i_2, i_6, i_4])\wedge$$

$$(\bigwedge_{0 \le q \le k-1} \psi[q])$$

is satisfiable if and only if there exists a reconfiguration plan for C-Graphs I and G with at most k reconfiguration steps.

Note that

$$(\alpha = \beta) \Leftrightarrow ((\alpha \vee \neg\beta) \wedge (\neg\alpha \vee \beta)).$$

Therefore, $\psi[q] \Leftrightarrow \psi'[q]$, where

$$\psi'[q] \rightleftharpoons \bigvee_{\substack{1 \le s_1 \le n,\\ 1 \le s_2 \le p_{i_1},\\ 1 \le s_3 \le n,\\ 1 \le s_4 \le p_{i_3}}} \bigwedge_{\substack{1 \le i_1 \le n,\\ 1 \le i_2 \le p_{i_1},\\ 1 \le i_3 \le n,\\ 1 \le i_4 \le p_{i_3},\\ i_1 \ne s_1,\\ i_2 \ne s_2,\\ i_3 \ne s_3,\\ i_4 \ne s_4,\\ i_1 \ne s_3,\\ i_2 \ne s_4,\\ i_3 \ne s_1,\\ i_4 \ne s_2}} ((x[q, i_1, i_2, i_3, i_4] \vee$$

$$\neg x[q+1, i_1, i_2, i_3, i_4])\wedge$$

$$(\neg x[q,i_1,i_2,i_3,i_4] \vee$$

$$x[q+1,i_1,i_2,i_3,i_4])),$$

$$\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3}}} \bigwedge_{j=1}^{n^2} w[j,i_1,i_2,i_3,i_4] =$$

$$w[j,i_3,i_2,i_1,i_4] \Leftrightarrow$$

$$\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3}}} \bigwedge_{j=1}^{n^2} ((w[j,i_1,i_2,i_3,i_4] \vee$$

$$\neg w[j,i_3,i_2,i_1,i_4]) \wedge$$

$$(\neg w[j,i_1,i_2,i_3,i_4] \vee$$

$$w[j,i_3,i_2,i_1,i_4])).$$

Since $\alpha \to \beta \Leftrightarrow \neg\alpha \vee \beta$,

$$\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le n}} w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4] \to$$

$$x[k,i_1,i_2,i_3,i_4] = z[k,i_5,i_2,i_6,i_4] \Leftrightarrow$$

$$\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le n}} \neg w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4] \vee$$

$$x[k,i_1,i_2,i_3,i_4] = z[k,i_5,i_2,i_6,i_4] \Leftrightarrow$$

$$\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le n}} (\neg w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4] \vee$$

$$((x[k,i_1,i_2,i_3,i_4] \vee$$

$$\neg z[k,i_5,i_2,i_6,i_4]) \wedge$$

$$(\neg x[k,i_1,i_2,i_3,i_4] \vee$$

$$z[k,i_5,i_2,i_6,i_4]))).$$

So, using only $\neg$, $\wedge$, and $\vee$, we obtain a boolean function

$$\xi_1 \rightleftharpoons (\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\in E[0]} x[0,i_1,i_2,i_3,i_4]) \wedge$$

$$(\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\notin E[0]} \neg x[0,i_1,i_2,i_3,i_4]) \wedge$$

$$(\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\in E[k]} z[k,i_1,i_2,i_3,i_4]) \wedge$$

$$(\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\notin E[k]} \neg z[k,i_1,i_2,i_3,i_4]) \wedge$$

$$(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},}} (\bigvee_{j=1}^{n^2} w[j,i_1,i_2,i_3,i_4])) \wedge$$

$$(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},}} (\bigwedge_{\substack{1\le j_1\le n^{n^2},\\ 1\le j_2\le n^{n^2},\\ j_1\ne j_2}} (\neg w[j_1,i_1,i_2,i_3,i_4] \vee$$

$$\neg w[j_2,i_1,i_2,i_3,i_4]))) \wedge$$

$$(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le p_{i_1},\\ 1\le i_7\le n,\\ 1\le i_8\le p_{i_3},\\ (i_1,i_3)\ne(i_5,i_7),\\ (i_1,i_3)\ne(i_7,i_5)}} \bigwedge_{j=1}^{n^2} (\neg w[j,i_1,i_2,i_3,i_4] \vee$$

$$\neg w[j,i_5,i_6,i_7,i_8])) \wedge$$

$$(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3}}} \bigwedge_{j=1}^{n^2} ((w[j,i_1,i_2,i_3,i_4] \vee$$

$$\neg w[j,i_3,i_2,i_1,i_4]) \wedge$$

$$(\neg w[j,i_1,i_2,i_3,i_4] \vee$$

$$w[j,i_3,i_2,i_1,i_4]))) \wedge$$

$$\Big(\bigwedge_{\substack{1\le i_1\le n,\\1\le i_2\le p_{i_1},\\1\le i_3\le n,\\1\le i_4\le p_{i_3},\\1\le i_5\le n,\\1\le i_6\le n}} (\neg w[(i_5-1)n+i_6, i_1, i_2, i_3, i_4]\vee ((x[k,i_1,i_2,i_3,i_4]\vee \neg z[k,i_5,i_2,i_6,i_4])\wedge (\neg x[k,i_1,i_2,i_3,i_4]\vee z[k,i_5,i_2,i_6,i_4]))))\wedge \Big(\bigwedge_{0\le q\le k-1} \psi'[q]\Big)$$

such that ξ_1 is satisfiable if and only if there exists a reconfiguration plan for C-Graphs I and G with at most k reconfiguration steps. It is easy to see that the size of boolean function ξ_1 polynomially depends from the size of C-Graphs. Therefore, we obtain an explicit reduction from ORP to PSAT.

Clearly, ξ_1 is not in conjunctive normal form. Using the distributive law, we can obtain from ξ_1 a boolean function in conjunctive normal form but this function will be have exponential size. In some sense it is a good news. The propositional satisfiability problem seems to become easier if boolean functions are restricted to those in disjunctive normal form. This is because such a formula is satisfiable if and only if some clause is satisfiable, and a conjunctive clause is satisfiable if and only if it does not contain both x and $\neg x$ for some variable x. This can be checked in polynomial time. Correspondently, the propositional satisfiability problem seems to become harder if boolean functions are restricted to those in conjunctive normal form. From this point of view the impossibility of polynomial reduction from ξ_1 to a boolean function in conjunctive normal form is a good news.

B. Reduction to SAT

It is easy to see that $\psi[q] \Leftrightarrow \psi''[q]$, where

$$\psi''[q] \rightleftharpoons \bigwedge_{\substack{1\le s_1\le n,\\1\le s_2\le p_{i_1},\\1\le s_3\le n,\\1\le s_4\le p_{i_3}\\1\le t_1\le n,\\1\le t_2\le p_{i_1},\\1\le t_3\le n,\\1\le t_4\le p_{i_3},\\(s_1,s_2,s_3,s_4)\ne(t_1,t_2,t_3,t_4),\\(s_1,s_2,s_3,s_4)\ne(t_3,t_4,t_1,t_2),}} (x[q,s_1,s_2,s_3,s_4] = x[q+1,s_1,s_2,s_3,s_4]\vee x[q,t_1,t_2,t_3,t_4] = x[q+1,t_1,t_2,t_3,t_4]).$$

Since

$$(\alpha=\beta)\Leftrightarrow((\alpha\vee\neg\beta)\wedge(\neg\alpha\vee\beta)),$$

it is clear that $\psi''[q]\Leftrightarrow\psi'''[q]$, where

$$\psi'''[q] \rightleftharpoons \bigwedge_{\substack{1\le s_1\le n,\\1\le s_2\le p_{i_1},\\1\le s_3\le n,\\1\le s_4\le p_{i_3}\\1\le t_1\le n,\\1\le t_2\le p_{i_1},\\1\le t_3\le n,\\1\le t_4\le p_{i_3},\\(s_1,s_2,s_3,s_4)\ne(t_1,t_2,t_3,t_4),\\(s_1,s_2,s_3,s_4)\ne(t_3,t_4,t_1,t_2),}} ((x[q,s_1,s_2,s_3,s_4]\vee \neg x[q+1,s_1,s_2,s_3,s_4])\wedge (\neg x[q,s_1,s_2,s_3,s_4]\vee x[q+1,s_1,s_2,s_3,s_4]))\vee ((x[q,t_1,t_2,t_3,t_4]\vee \neg x[q+1,t_1,t_2,t_3,t_4])\wedge (\neg x[q,t_1,t_2,t_3,t_4]\vee x[q+1,t_1,t_2,t_3,t_4])).$$

Note that

$$((x[q,s_1,s_2,s_3,s_4]\vee \neg x[q+1,s_1,s_2,s_3,s_4])\wedge (\neg x[q,s_1,s_2,s_3,s_4]\vee x[q+1,s_1,s_2,s_3,s_4]))\vee ((x[q,t_1,t_2,t_3,t_4]\vee \neg x[q+1,t_1,t_2,t_3,t_4])\wedge (\neg x[q,t_1,t_2,t_3,t_4]\vee x[q+1,t_1,t_2,t_3,t_4]))\Leftrightarrow$$
$$(((x[q,s_1,s_2,s_3,s_4]\vee \neg x[q+1,s_1,s_2,s_3,s_4])\wedge (\neg x[q,s_1,s_2,s_3,s_4]\vee x[q+1,s_1,s_2,s_3,s_4]))\vee (x[q,t_1,t_2,t_3,t_4]\vee \neg x[q+1,t_1,t_2,t_3,t_4]))\wedge (((x[q,s_1,s_2,s_3,s_4]\vee \neg x[q+1,s_1,s_2,s_3,s_4])\wedge (\neg x[q,s_1,s_2,s_3,s_4]\vee x[q+1,s_1,s_2,s_3,s_4]))\vee$$

$$(\neg x[q,t_1,t_2,t_3,t_4]\vee$$
$$x[q+1,t_1,t_2,t_3,t_4])) \Leftrightarrow$$
$$(x[q,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q+1,s_1,s_2,s_3,s_4]\vee$$
$$x[q,t_1,t_2,t_3,t_4]\vee$$
$$\neg x[q+1,t_1,t_2,t_3,t_4])\wedge$$
$$(\neg x[q,s_1,s_2,s_3,s_4]\vee$$
$$x[q+1,s_1,s_2,s_3,s_4]\vee$$
$$x[q,t_1,t_2,t_3,t_4]\vee$$
$$\neg x[q+1,t_1,t_2,t_3,t_4])\wedge$$
$$(x[q,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q+1,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q,t_1,t_2,t_3,t_4]\vee$$
$$x[q+1,t_1,t_2,t_3,t_4])\wedge$$
$$(\neg x[q,s_1,s_2,s_3,s_4]\vee$$
$$x[q+1,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q,t_1,t_2,t_3,t_4]\vee$$
$$x[q+1,t_1,t_2,t_3,t_4]).$$

Therefore, $\psi'''[q] \Leftrightarrow \psi''''[q]$, where

$$\psi''''[q] \rightleftharpoons \bigwedge_{\substack{1\le s_1\le n,\\ 1\le s_2\le p_{i_1},\\ 1\le s_3\le n,\\ 1\le s_4\le p_{i_3}\\ 1\le t_1\le n,\\ 1\le t_2\le p_{i_1},\\ 1\le t_3\le n,\\ 1\le t_4\le p_{i_3},\\ (s_1,s_2,s_3,s_4)\ne(t_1,t_2,t_3,t_4),\\ (s_1,s_2,s_3,s_4)\ne(t_3,t_4,t_1,t_2),}} (x[q,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q+1,s_1,s_2,s_3,s_4]\vee x[q,t_1,t_2,t_3,t_4]\vee$$
$$\neg x[q+1,t_1,t_2,t_3,t_4])\wedge$$
$$(\neg x[q,s_1,s_2,s_3,s_4]\vee$$
$$x[q+1,s_1,s_2,s_3,s_4]\vee$$
$$x[q,t_1,t_2,t_3,t_4]\vee$$
$$\neg x[q+1,t_1,t_2,t_3,t_4])\wedge$$
$$(x[q,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q+1,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q,t_1,t_2,t_3,t_4]\vee$$
$$x[q+1,t_1,t_2,t_3,t_4])\wedge$$
$$(\neg x[q,s_1,s_2,s_3,s_4]\vee$$
$$x[q+1,s_1,s_2,s_3,s_4]\vee$$
$$\neg x[q,t_1,t_2,t_3,t_4]\vee$$
$$x[q+1,t_1,t_2,t_3,t_4]).$$

Note that

$$\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le n}} (\neg w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4]\vee$$
$$((x[k,i_1,i_2,i_3,i_4]\vee$$
$$\neg z[k,i_5,i_2,i_6,i_4])\wedge$$
$$(\neg x[k,i_1,i_2,i_3,i_4]\vee$$
$$z[k,i_5,i_2,i_6,i_4]))) \Leftrightarrow$$
$$\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le n}} ((\neg w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4]\vee$$
$$x[k,i_1,i_2,i_3,i_4]\vee$$
$$\neg z[k,i_5,i_2,i_6,i_4])\wedge$$
$$(\neg w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4]\vee$$
$$\neg x[k,i_1,i_2,i_3,i_4]\vee$$
$$z[k,i_5,i_2,i_6,i_4])).$$

So, we obtain a boolean function

$$\xi_2 \rightleftharpoons (\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\in E[0]} x[0,i_1,i_2,i_3,i_4])\wedge$$
$$(\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\notin E[0]} \neg x[0,i_1,i_2,i_3,i_4])\wedge$$
$$(\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\in E[k]} z[k,i_1,i_2,i_3,i_4])\wedge$$
$$(\bigwedge_{(v[i_1,i_2],v[i_3,i_4])\notin E[k]} \neg z[k,i_1,i_2,i_3,i_4])\wedge$$
$$(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},}} (\bigvee_{j=1}^{n^2} w[j,i_1,i_2,i_3,i_4]))\wedge$$

$$
\Big(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},}} \Big(\bigwedge_{\substack{1\le j_1\le n^{n^2},\\ 1\le j_2\le n^{n^2},\\ j_1\ne j_2}} (\neg w[j_1,i_1,i_2,i_3,i_4]\vee \neg w[j_2,i_1,i_2,i_3,i_4])\Big)\Big)\wedge
$$

$$
\Big(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le p_{i_1},\\ 1\le i_7\le n,\\ 1\le i_8\le p_{i_3},\\ (i_1,i_3)\ne(i_5,i_7),\\ (i_1,i_3)\ne(i_7,i_5)}} \bigwedge_{j=1}^{n^2} (\neg w[j,i_1,i_2,i_3,i_4]\vee \neg w[j,i_5,i_6,i_7,i_8])\Big)\wedge
$$

$$
\Big(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3}}} \bigwedge_{j=1}^{n^2} ((w[j,i_1,i_2,i_3,i_4]\vee \neg w[j,i_3,i_2,i_1,i_4])\wedge (\neg w[j,i_1,i_2,i_3,i_4]\vee w[j,i_3,i_2,i_1,i_4]))\Big)\wedge
$$

$$
\Big(\bigwedge_{\substack{1\le i_1\le n,\\ 1\le i_2\le p_{i_1},\\ 1\le i_3\le n,\\ 1\le i_4\le p_{i_3},\\ 1\le i_5\le n,\\ 1\le i_6\le n}} ((\neg w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4]\vee x[k,i_1,i_2,i_3,i_4]\vee \neg z[k,i_5,i_2,i_6,i_4])\wedge (\neg w[(i_5-1)n+i_6,i_1,i_2,i_3,i_4]\vee \neg x[k,i_1,i_2,i_3,i_4]\vee z[k,i_5,i_2,i_6,i_4]))\Big)\wedge
$$

$$
\Big(\bigwedge_{0\le q\le k-1} \psi''''[q]\Big)
$$

such that ξ_2 is satisfiable if and only if there exists a reconfiguration plan for C-Graphs I and G with at most k reconfiguration steps. It is easy to see that the size of boolean function ξ_2 polynomially depends from the size of C-Graphs. Since ξ_2 in conjunctive normal form, we obtain an explicit reduction from ORP to SAT.

IV. Conclusion and Experimental Results

In recent years, many optimization methods, parallel algorithms, and practical techniques have been developed for solving the satisfiability problem (see [50]). In particular, proposed several genetic algorithms [51] – [54]. Considered hybrid algorithms in which the approach of genetic algorithms combined with local search [55].

Modern propositional satisfiability solvers are usually designed to solve SAT formula encoded in conjunctive normal form (CNF). Stochastic local search techniques have been successful in solving propositional satisfiability problems encoded in CNF. Recently complete solvers have shown that there are advantages to tackling propositional satisfiability problems in a more expressive natural representation, since the conversion to CNF can lose problem structure and introduce significantly more variables to encode the problem. CNF solvers can be disadvantageous for problems which are more naturally encoded as arbitrary propositional formula. The conversion to CNF form may increase the size of the formula exponentially, or significantly reduce the strength of the formulation. The translation may introduce many new variables which increases the size of the raw valuation space through which the solver must search. Recently, interest has arisen in designing non-clausal satisfiability algorithms (see e.g. [56] – [63]).

Relatively high efficiency demonstrated by algorithms based solely on local search. Of course, these algorithms require exponential time at worst. But they can relatively quick receive solutions for many boolean functions. Therefore, it is natural to use a reduction to different variants of the satisfiability problem to solve computational hard problems.

Encoding problems as Boolean satisfiability and solving them with very efficient satisfiability algorithms has recently caused considerable interest. In particular, local search algorithms have given impressive results on many problems. For example, there are several ways of SAT-encoding constraint satisfaction [64] – [73], clique [74], planning [75] – [95], and colouring problems [74], [96] – [98]. The maximum cut, vertex cover and maximum independent set problems can be reduced to MAX-2-SAT [99] – [101]. There are a number of implicit reductions from the Hamiltonian cycle problem to the satisfiability (SAT) problem (see [74], [102], [103]).

In previous section we obtain an implicit reduction from the optimal reconfiguration planning problem of finding the least number of reconfiguration steps to transform between two configurations to some variants of satisfiability: PSAT, SAT. We create a generator of special hard and natural instances for the optimal reconfiguration planning problem of finding the least number of reconfiguration steps to transform between two configurations. We use algorithms from [104]. Also we design our own genetic algorithm for SAT which based on algorithms from [104]. We use heterogeneous cluster based on three clusters (Cluster USU, Linux, 8 calculation nodes, Intel Pentium IV 2.40GHz processors; umt, Linux, 256 calculation nodes, Xeon 3.00GHz processors; um64, Linux, 124 calculation nodes, AMD Opteron 2.6GHz bi-processors)

[105]. For computational experiment we create special hard test sets and natural test sets. Special hard test sets based on ideas from [106]. Natural test sets based on ideas from [48]. In tests we consider systems consisted from approximately 400 of modular robots.

Each test was run on a cluster of at least 100 nodes. For special hard test sets: the maximum solution time was 16 hours; the average time to find a solution was 33.2 minutes; the best time was 116 seconds. For natural test sets: the maximum solution time was 9 hours; the average time to find a solution was 9.8 minutes; the best time was 9 seconds. Based on our experiments we can say that considered model can be used as an efficient planner.

Acknowledgment

The work was partially supported by Grant of President of the Russian Federation MD-1687.2008.9 and Analytical Departmental Program "Developing the scientific potential of high school" 2.1.1/1775.

References

[1] Rus, D., and Chirikjian, G., Eds., *Special Issue on Self-Reconfiguring Robots*, Autonomous Robotics, 2001, 10(1).

[2] Chirikjian, G.S. Kinematics of a Metamorphic Robotic System. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 449–455, San Diego, 1994. IEEE Computer Society Press.

[3] Fukuda, T., and Nakagawa, S. A Dynamically Reconfigurable Robotic System (Concept of a System and Optimal Configurations). *Proceedings of the 1987 IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, pages 588–595, Los Alamitos, 1987. IEEE Computer Society Press.

[4] Hosokawa, K., Tsujimori, T., Fujii, T., Kaetsu, H., Asama, H., Kuroda, Y., and Endo, I. Self-Organizing Collective Robots with Morphogenesis in a Vertical Plane. *Proceedings of the 1998 IEEE International Conference on Robotics and Automation. Vol. 4*, pages 2858–2863, Leuven , 1998. IEEE Computer Society Press.

[5] Murata, S., Kurokawa, H., and Kokaji, S. Self-Assembling Machine. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 442–448, San Diego, 1994. IEEE Computer Society Press.

[6] Pamecha, A., Chiang, C., Stein, D., and Chirikjian, G. Design and implementation of metamorphic robots. *Proceedings of The 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, pages 1–10, Irvine, 1996. ASME Press.

[7] Rus, D., and Vona, M. Crystalline Robots: Self-reconfiguration with Unit-compressible Modules. *Autonomous Robots*, 10(1):107–124, 2001.

[8] Kotay, K., and Rus, D. Locomotion versatility through selfreconfiguration. *Robotics and Autonomous Systems*, 26(2-3):217–232, 1999.

[9] Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., and Kokaji, S. M-TRAN: Self-Reconfigurable Modular Robotic System. *IEEE/ASME Transactions on Mechatronics*, 7(4):431–441, 2002.

[10] Suh, J., Homans, S., and Yim, M. Telecubes: Mechanical Design of a Module for Self-Reconfiguring Robotics. *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 4095–4101, Washington, 2002. IEEE Computer Society Press.

[11] Benhabib, B., Zak, G., and Lipton, M.G. A Generalized Kinematic Modeling Method for Modular Robots. *Journal of Robotic Systems*, 6(5):545–571, 1989.

[12] Cohen, R., Lipton, M.G., Dai, M.Q., and Benhabib, B. Conceptual Design of a Modular Robot. *ASME Journal of Mechanical Design*, 114:117–125, 1992.

[13] Sciaky, M. Modular Robots Implementation. In Nof, S., editor, *Handbook of Industrial Robotics*, pages 759–774. John Wiley and Sons, 1985.

[14] Wurst, K.H. The Conception and Construction of a Modular Robot System. In Van Brussel, H., editor, *16th International Syposium On Industrial Robots. 8th International Conference On Industrial Robot Technology: Proceedings*, pages 37–44, Brussels, 1986. Springer-Verlag.

[15] Beni, G. Concept of Cellular Robotic Systems. *Proceedings of the IEEE International Symposium on Intelligent Control*, pages 57–62, Arlington, 1988. IEEE Computer Society Press.

[16] Beni, G., and Wang, J. Theoretical Problems for the Realization of Distributed Robotic Systems. *Proceedings of the 1991 IEEE Conference on Robotics and Automation*, pages 1914–1920, Sacramento, 1991. IEEE Computer Society Press.

[17] Fukuda, T., and Nakagawa, S. Dynamically Reconfigurable Robotic Systems. *Proceedings of the 1988 IEEE Conference on Robotics and Automation*, pages 1581–1586, Philadelphia, 1988. IEEE Computer Society Press.

[18] Fukuda, T., and Kawauchi, Y. Cellular Robotic System (CEBOT) as One of the Realization of Self-organizing Intelligent Universal Manipulator. *Proceedings of the 1990 IEEE Conference on Robotics and Automation*, pages 662–667, Cincinnati, 1990. IEEE Computer Society Press.

[19] Beni, G., and Hackwood, S. Stationary Waves in Cyclic Swarms. *Proceedings of IEEE International Symposium on Intelligent Control*, pages 234–242, Los Alamitos, 1992. IEEE Computer Society Press.

[20] Beni, G., and Wang, J. Swarm Intelligence. *Proceedings of Seventh Annual Meeting of the Robotics Society of Japan*, pages 425–428, Tokyo, 1989. RSJ Press.

[21] Hackwood, S., and Beni, G. Self-organizing Sensors by Deterministic Annealing. *Proceedings of IEEE/RSJ International Conference on Intelligent Robot and Systems — IROS'91*, pages 1177–1183, Los Alamitos, 1991. IEEE Computer Society Press.

[22] Hackwood, S., and Beni, G. Self-organization of Sensors for Swarm Intelligence. *Proceedings of IEEE International Conference on Robotics and Automation*, pages 819–829, Nice, 1992. IEEE Computer Society Press.

[23] Yim, M. A Reconfigurable Modular Robot with Many Modes of Locomotion. *Proceedings of the 1993 JSME International Conference on Advanced Mechatronics*, pages 283-288, Tokyo, 1993. JSME Press.

[24] Yim, M. Locomotion with a unit-modular reconfigurable robot. PhD thesis, Department of Mechanical Engineering, Stanford University, Stanford, 1994.

[25] Yim, M. New Locomotion Gaits. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2508–2524, San Diego, 1994. IEEE Computer Society Press

[26] Murata, S., Kurokawa, H., and Kokaji, S. Self-Organizing Machine. Video Proceedings, 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, May 1995.

[27] M. Ashley-Rollman, S. Goldstein, P. Lee, T. Mowry, and P. Pillai. *Meld: A declarative approach to programming ensembles*, Proceedings of the IEEE International Conference on Robots and Systems, 2007. pp.2794–2800.

[28] Charron-Bost, B., Delporte-Gallet, C., and Fauconnier, H. *Local and temporal predicates in distributed systems*, ACM Transactions on Programming Languages and Systems, 17(1):157–179, 1995.

[29] DeRosa, M., Goldstein, S., Lee, P., Campbell, J., and Pillai, P. *Scalable shape sculpting via hole motion: Motion planning in lattice constrained modular robots*, Proceedings of the IEEE International Conference on Robotics and Automation, 2006. pp.1462–1468.

[30] Dewey, D., and Srinivasa, S.S. *A planning framework for local metamorphic systems*, Technical Report CMU-RI-TR-XX, The Robotics Institute, Carnegie Mellon University, 2007.

[31] Ashley-Rollman, M.P., De Rosa, M., Srinivasa, S.S., Pillai, P., Goldstein, S.C., Campbell, J. *Declarative Programming for Modular Robots*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Self-Reconfigurable Robots and Systems and Applications, 2007. pp.1–6.

[32] Pamecha, A., Ebert-Uphoff, I., and Chirikjian, G. Useful metrics for modular robot motion planning. *IEEE Transactions on Robotics and Automation*, 13(4):531–545, 1997.

[33] Vassilvitskii, S., Kubica, J., Rieffel, E.G., Suh, J.W., and Yim, M. On the General Reconfiguration Problem for Expanding Cube Style Modular Robots. *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 801–808, Washington, 2002. IEEE Computer Society Press.

[34] Kurokawa, H., Tomita, K., Kamimura, A., Yoshida, E., Kokaji, S., and Murata, S. Distributed Self-reconfiguration Control of Modular Robot M-TRAN. *Proceedings of the Twenty-Seventh Annual International Conference on Cement Microscopy*, pages 254–259, Victoria, 2005. IEEE Computer Society Press.

[35] Hosokawa, K., Fujii, T., Kaetsu, H., Asama, H., Kuroda, Y., and Endo, I. Self-organizing collective robots with morphogenesis in a vertical plane. *JSME International Journal Series C Mechanical Systems Machine Elements and Manufacturing*, 42:195–202, 1999.

[36] Butler, Z., Murata, S., and Rus, D. Distributed Replication Algorithms for Self-Reconfiguring Modular Robots. *Proceedings of 6th International Symposium on Distributed Autanomous Robotic Systems (DARS'02)*, pages 25–27, Fukuda, 2002.
http://groups.csail.mit.edu/drl/wiki/images/c/c5/dars02.pdf

[37] Walter, J., Tsai, E., and Amato, N. Algorithms for Fast Concurrent Reconfiguration of Hexagonal Metamorphic Robots. *IEEE Transactions on Robotics*, 21(4):621–631, 2005.

[38] Ünsal, C., and Khosla, P.K. A Multi-layered Planner for Self-Reconfiguration of a Uniform Group of I-Cube Modules. *IEEE International Conference on intelligent Robots and Systems. Vol. 1*, pages 598–605, Maui, 2001. IEEE Computer Society Press.

[39] Reif, J.H., and Slee, S. Optimal Kinodynamic Motion Planning for 2D Reconfiguration of Self-Reconfigurable Robots. Robotics: Science and Systems, Conference, Georgia Institute of Technology, Atlanta, GA, June 27-30, 2007.
http://www.roboticsproceedings.org/rss03/p20.pdf

[40] Aloupis, G., Collette, S., Demaine, E.D., Langerman, S., Sacristán, V., and Wuhrer, S. Reconfiguration of Cube-Style Modular Robots Using O(log n) Parallel Moves. In Hong, S.H., Nagamochi, H., Fukunaga, T., editors, *Proceedings of the 19th Annual International Symposium on Algorithms and Computation — ISAAC 2008*, pages 342–353, Gold Coast, 2008. Springer-Verlag.

[41] Christensen, D., Ostergaard, E., and Lund, H.H. *Metamodule control for the atron self-reconfigurable robotic system*, Proceedings of the The 8th Conference on Intelligent Autonomous Systems, Amsterdam, 2004. pp.685–692.

[42] Dewey, D., Srinivasa, S.S., Ashley-Rollman, M.P., De Rosa, M., Pillai, P., Mowry, T.C., Campbell, J.D., and Goldstein, S.C. *Generalizing Metamodules to Simplify Planning in Modular Robotic Systems*, Proceedings of IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, 2008. pp.1338–1345.

[43] Casal, A., and Yim, M. Self-Reconfiguration Planning For a Class of Modular Robots. *Proceedings of SPIE. Vol. 3839. Sensor Fusion and Decentralized Control in Robotic Systems. Vol. II*, pages 246–257, Boston, 1999. SPIE Press.

[44] Nelson, C.A. A framework for self-reconfiguration planning for unit-modular robots. Phd Thesis, Purdue University, Department of Mechanical Engineering, Purdue, 2005.

[45] Gay, S. Roombots: Toward Emancipation of Furniture. A Kinematics-Dependent Reconfiguration Algorithm for Chain-Type Modular Robots. Master Thesis, Ecole Polytechnique, Department of Computer Science, Ecole, 2007.

[46] Shen, W.-M., Salemi, B., and Will, P. Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots. *IEEE Transactions on Robotics and Automation*, 18(5):700–712, 2002.

[47] Hou, F., and Shen, W.-M. Distributed, Dynamic, and Autonomous Reconfiguration Planning for Chain-Type Self-Reconfigurable Robots. *Proceedings of 2008 IEEE International Conference on Robotics and Automation*, pages 3135–3140, Pasadena, 2008. IEEE Computer Society Press.

[48] Hou, F., Shen, W.-M. On the Complexity of Optimal Reconfiguration Planning for Modular Reconfigurable Robots. *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, pages 2791–2796, Anchorage, 2010. IEEE Computer Society Press.

[49] Cook, S.A. The Complexity of Theorem Proving Procedures. In Harrison, M.A., Banerji, R.B., Ullman, J.D., editors, *STOC '71 Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, 1971. Association for Computing Machinery Press.

[50] Gu, J., Purdom, P., Franco, J., and Wah, B. Algorithms for the Satisfiability (SAT) Problem: A Survey. In Johnson, David, and Trick, Michael, editors, *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, pages 19–152. American Mathematical Society, Providence, Rhode Island, 1996.

[51] Fleurent, J. Genetic algorithms and hybrids for graph coloring. *Annals of Operations Research*, 63(3):437–461, 1996.

[52] Hao, J., and Dorne, R. A new population-based method for satisfiability problems. In Cohn, A.G., editor, *Proceedings of 11th European Conference on Artificial Intelligence*, pages 135–139, Amsterdam, 1994. John Wiley & Sons.

[53] Jong, K., and Spears, W. Using genetic algorithms to solve np-complete problems.
In Schaffer, J.D., editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 124-132, Fairfax, 1989. Morgan Kaufmann Publishers Inc.

[54] Voorn, R., Dastani, M., and Marchiori, E. Finding simplest pattern structures using genetic programming. In Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 3–10, San Francisco, 2001. Morgan Kaufmann Publishers Inc.

[55] Hao, J., Lardeux, F., and Saubion, F. A hybrid genetic algorithm for the satisfiability problem. *Proceedings of the 1rst International Workshop on Heuristics*, pages 102–109, Beijing, 2002. Springer-Verlag.

[56] Armando, A., and Giunchiglia, E. Embedding complex decision procedures inside an interactive theorem prover. *Annals of Mathematics and Artificial Intelligence*, 8(3-4):475–502, 1993.

[57] Giunchiglia, E., and Sebastiani, R. Applying the Davis-Putnam procedure to nonclausal formulas. In Lamma, E., Mello, P., editors, *AI∗IA 99: Advances in Artificial Intelligence, 6th Congress of the Italian Association for Artificial Intelligence*, pages 84–94, Bologna, 2000. Springer-Verlag.

[58] Kautz, H., Selman, B., and McAllester, D. Exploiting variable dependency in local search. *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 7–9, Nagoya, 1997. Morgan Kaufmann Publishers Inc.

[59] Muhammad, R., and Stuckey, P.J. A Stochastic Non-CNF SAT Solver. In Yang, Q., Webb, G.I., editors, *PRICAI 2006: Trends in Artificial Intelligence, 9th Pacific Rim International Conference on Artificial Intelligence*, pages 120–129, Guilin, 2006. Springer-Verlag.

[60] Sebastiani, R. Applying GSAT to non-clausal formulas. *Journal of Artificial Intelligence Research*, 1:309–314, 1994.

[61] Selman, B., Kautz, H., and Cohen, B. Noise strategies for improving local search. In Hayes-Roth, B., Korf, R.E., editors, *AAAI'94 Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 337–343, Seattle, 1994. American Association for Artificial Intelligence.

[62] Stachniak, Z. Going non-clausal. *5th International Symposium on Theory and Applications of Satisfiability Testing: SAT 2002*, pages 316–322, Cincinnati, 2002. Springer-Verlag.

[63] Thiffault, C., Bacchus, F., and Walsh, T. Solving non-clausal formulas with DPLL search. In Hoos, H.H., Mitchell, D.G., editors, *Theory and Applications of Satisfiability Testing: 7th International Conference: SAT 2004*, pages 663–678, Vancouver, 2004. Springer-Verlag.

[64] Bessiere, C., Hebrard, E., and Walsh, T. Local Consistencies in SAT. *Theory and applications of satisfiability testing: SAT 2003: international conference on theory and applications of satisfiability testing N 6*, pages 400–407, Santa Margherita Ligure , 2003. Springer-Verlag.

[65] Davis, M., Logemann, G., and Loveland, D. A Machine Program for Theorem Proving. *Communications of the ACM*, 5(7):394–397, 1962.

[66] Frisch, A., and Peugniez, T. Solving Non-Boolean Satisfiability Problems with Stochastic Local Search. In Nebel, B., editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 282–288, Seattle, 2001. pp. 282 - 288. Morgan Kaufmann Publishers Inc.

[67] Frisch, A.M., Peugniez, T.J., Doggett, A.J., and Nightingale, P.W. Solving Non-Boolean Satisfiability Problems with Stochastic Local Search: A Comparison of Encodings. *Journal of Automated Reasoning*, 35(1-3):143–179, 2005.

[68] Genisson, R., and Jegou, P. Davis and Putnam Were Already Forward Checking. In Wahlster, W., editor, *Twelfth European Conference on Artificial Intelligence*, pages 180–184, Budapest, 1996. John Wiley and Sons.

[69] Gent, I. Arc Consistency in SAT. In van Harmelen, F., editor, *Proceedings of the Fifteenth European Conference on Artificial Intelligence*, pages 121–125, Lyons, 2002. IOS Press.

[70] Kasif, S. On the Parallel Complexity of Discrete Relaxation in Constraint Satisfaction Networks. *Artificial Intelligence*, 45(3):275–286, 1990.

[71] Prestwich, S.D. Local search on SAT-encoded colouring problems. *Theory and applications of satisfiability testing: SAT 2003: international conference on theory and applications of satisfiability testing N 6*, pages 105–119, Santa Margherita Ligure , 2003. Springer-Verlag.

[72] Sabin, D., and Freuder, G. Contradicting Conventional Wisdom in Constraint Satisfaction. In Cohn, A.G., editor, *Proceedings of the Eleventh*

European Conference on Artificial Intelligence, pages 125–129, Amsterdam, 1994. John Wiley and Sons.

[73] Walsh, T. SAT v CSP. In Dechter, R., editor, *Sixth International Conference on Principles and Practice of Constraint Programming*, pages 441–456, Singapore, 2000. Springer-Verlag.

[74] Iwama, K., and Miyazaki, S. SAR-variable complexity of hard combinatorial problems. *IFIP Transactions A: Computer Science and Technology*, 1:253–258, 1994.

[75] Büttner, M., and Rintanen, J. Improving parallel planning with constraints on the number of operators. In Biundo, S., Myers, K., and Rajan, K., editors, *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling*, pages 292–299, Monterey, 2005. AAAI Press.

[76] Ernst, M., Millstein, T., and Weld, D. Automatic SAT-Compilation of Planning Problems. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 1169–1176, Nagoya, 1997. Morgan Kaufmann Publishers Inc.

[77] Kautz, H. SATPLAN04: Planning as Satisfiability. In Edelkamp, S., Hoffmann, J., Littman, M., and Younes, H., editors, *Proceedings of the 4th International Planning Competition at the 14th International Conference on Automated Planning and Scheduling*, pages 44–45, Whistler, 2004. AAAI Press.

[78] Kautz, H., McAllester, D., and Selman, B. Encoding Plans in Propositional Logic. In Aiello, L.C., Doyle, J., Shapiro, S.C., editors, *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning*, pages 374–384, Cambridge, Massachusetts, 1996. Morgan Kaufmann Publishers Inc.

[79] Kautz, H., and Selman, B. Planning as Satisfiability. In Neumann, B., editor, *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 359–363, Vienna, 1992. John Wiley & Sons.

[80] Kautz, H., and Selman, B. Pushing the envelope: planning, propositional logic, and stochastic search. In Brewka, G., editor, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Annual Conference on Innovative Applications of Artificial Intelligence*, pages 1194–1201, Portland, 1996. AAAI Press.

[81] Kautz, H., and Selman, B. Unifying SAT-based and graph-based planning. In Dean, T., editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 318–325, Stockholm, 1999. Morgan Kaufmann Publishers Inc.

[82] Mattmüller, R., and Rintanen, J. Planning for temporally extended goals as propositional satisfiability. In Veloso, M., editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1966–1971, Hyderabad, 2007. AAAI Press.

[83] Rintanen, J. Compact representation of sets of binary constraints. In Perini, A., Penserini, L., and Peppas, P., editors, *Proceedings of the 17th European Conference on Artificial Intelligence*, pages 143–147, Trento, 2006. IOS Press.

[84] Rintanen, J. Evaluation strategies for planning as satisfiability. In Lopez de Mantaras, R., and Saitta, L., editors, *ECAI 2004: Proceedings of the 16th European Conference on Artificial Intelligence*, pages 682–687, Valencia, 2004. IOS Press.

[85] Rintanen, J. Heuristic Planning with SAT: Beyond Uninformed Depth-First Search. In Li, J., editor, *AI 2010: Advances in Artificial Intelligence*, pages 415–424, Adelaide, 2010. Springer-Verlag.

[86] Rintanen, J. Heuristics for Planning with SAT. In Cohen, D., editor, *Principles and Practice of Constraint Programming: 16th International Conference*, pages 414–428, St. Andrews, 2010. Springer-Verlag.

[87] Rintanen, J. Planning graphs and propositional clause-learning. In Brewka, G., and Doherty, P., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference*, pages 535–543, Sydney, 2008. AAAI Press.

[88] Rintanen, J. Symmetry reduction for SAT representations of transition systems. In Giunchiglia, E., Muscettola, N., and Nau, D., editors, *Proceedings of the 13th International Conference on Automated Planning and Scheduling*, pages 32–40, Trento, 2003. AAAI Press.

[89] Rintanen, J. A planning algorithm not based on directional search. In Cohn, A.G., Schubert, L.K., and Shapiro, S.C., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference*, pages 617–624, Trento, 1998. Morgan Kaufmann Publishers Inc.

[90] Rintanen, J. Partial implicit unfolding in the Davis-Putnam procedure for quantified Boolean formulae. In Nieuwenhuis, R., and Voronkov, A., editors, *International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, pages 362–376, Havana, 2001. Springer-Verlag.

[91] Rintanen, J., Heljanko, K., and Niemelä, I. Planning as Satisfiability: Parallel Plans and Algorithms for Plan Search. Technical Report 216, Institute of Computer Science, University of Freiburg, Freiburg, Germany, 2005.

[92] Rintanen, J., Heljanko, K., and Niemelä, I. Planning as satisfiability: parallel plans and algorithms for plan search. *Artificial Intelligence*, 170(12-13):1031–1080, 2006.

[93] Rintanen, J., Heljanko, K., and Niemelä, I. Parallel encodings of classical planning as satisfiability. In Alferes, J.J., and Leite, J., editors, *Logics in Artificial Intelligence: 9th European Conference*, pages 307–319, Lisbon, 2004. Springer-Verlag.

[94] Rintanen, J., and Jungholt, H. Numeric state variables in constraint-based planning. In Biundo, S., and Fox, M., editors, *Recent Advances in AI Planning: 5th European Conference on Planning*, pages 109–121, Durham, 2000. Springer-Verlag.

[95] Wehrle, M., and Rintanen, J. Planning as satisfiability with relaxed ∃-step plans. In Orgun, M., and Thornton, J., editors, *AI 2007 : Advances in Artificial Intelligence: 20th Australian Joint Conference on Artificial Intelligence*, pages 244–253, Surfers Paradise, Gold Coast, Australia, 2007. Springer-Verlag.

[96] van Gelder, A. Another Look at Graph Coloring via Propositional Satisfiability. In Mehrotra, A., Johnson, D.S., and Trick, M., editors, *Computational Symposium on Graph Coloring and its Generalizations*, pages 48–54, Ithaca, New York, 2002. Springer-Verlag.

[97] Bouhmala, N., Granmo, O.-C. Stochastic Learning for SAT- Encoded Graph Coloring Problems. *International Journal of Applied Metaheuristic Computing*, 1(3):1–19, 2010.

[98] Velev, M.N. Exploiting hierarchy and structure to efficiently solve graph coloring as SAT. In Gielen, G., editor, *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, pages 135–142, San Jose, 2007. IEEE Computer Society Press.

[99] Poljak, S., and Tuza, Z. Maximum cuts and largest bipartite subgraphs. In Cook, William, Lovasz, Laszlo, Seymour, Paul, editors, *Combinatorial Optimization*, pages 181–244. American Mathematical Society, Providence, Rhode Island, 1995.

[100] Cheriyan, J., Cunningnham, W.H., Tuncel, L., and Wang, Y. A linear programming and rounding approach to Max 2-Sat. In Johnson, David, and Trick, Michael, editors, *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, pages 395–414. American Mathematical Society, Providence, Rhode Island, 1996.

[101] Mahajan, M., and Raman, V. Parameterizing above guaranteed values: MaxSat and MaxCut. *Journal of Algorithms*, 31(2):335–354, 1999.

[102] Hoos, H.H. SAT-Encodings, Search Space Structure, and Local Search Performance. In Dean, T., editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 296–302, Stockholm, 1999. Morgan Kaufmann Publishers Inc.

[103] Plotnikov, A.D. A Logical Model of HCP. *International Journal of Mathematics and Mathematical Sciences*, 26(11), 2001.

[104] http://people.cs.ubc.ca/∼hoos/SATLIB/index-ubc.html

[105] http://parallel.imm.uran.ru/mvc_now/hardware/supercomp.htm

[106] Navarro, J.A., and Voronkov, A. Generation of Hard Non-Clausal Random Satisfiability Problems *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 436–442, Pittsburgh, 2005. AAAI Press.