Contemporary Engineering Sciences, Vol. 8, 2015, no. 19, 853 - 857
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2015.55159

The Permutation Problem Using a Unit-Capacity Robot

Anna Gorbenko

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia

Abstract

In this paper, we consider the permutation problem using a unit-capacity robot. In particular, we consider an approach to solve the problem. This approach is based on an explicit reduction from the problem to the satisfiability problem.

Subject Classification: 68T40

Keywords: unit-capacity robot, satisfiability problem, explicit reduction, genetic algorithms

The permutation problem using a unit-capacity robot (PPCR) arises in nuclear power plant fuel renewal (see [1, 2]). The PPCR can be considered as a particular pickup and delivery problem [3]. The PPCR is **NP**-complete [2]. In this paper, we consider an approach to solve PPCR. This approach is based on an explicit reduction from PPCR to the satisfiability problem.

In PPCR, it is assumed that the directed graph

$$G = (V, A)$$

describes all possible moves. We consider a set of

$$P = \{1, 2, \ldots, p\}$$

pieces that have to be moved over a set $W \subseteq V$ of particular locations. It is assumed that any location contains at most one piece.

Each piece from P possesses a specific type. For any type, there exists only one node in V having a piece of this type. We assume that, in the initial state, each piece is assigned to a given location and, in the final state, each location must contain a piece of a given type. Is there a sequence of $n \leq N$ states from the initial state to the final state?

Pieces are handled by a unit-capacity robot. It is assumed that such a robot can only move one piece at a time. Also, it is assumed that each piece can be moved only once.

We need two functions for initial and final states:

$$I : V \to P \cup \{0\},$$

$$F : V \to P \cup \{0\}.$$

For any j such that $1 \leq j \leq |V|$, let $I[j] = 0$ if and only if, in the initial state, node j containing no piece. Let $I[j] = k$, $0 < k \leq p$, $1 \leq j \leq |V|$, if and only if, in the initial state, node j having piece of type k. We assume that $F[j] = 0$, $1 \leq j \leq |V|$, if and only if, in the final state, node j containing no piece. Let $F[j] = k$, $0 < k \leq p$, $1 \leq j \leq |V|$, if and only if, in the final state, node j having piece of type k.

$$\varphi[1] = \bigwedge_{\substack{1 \leq j \leq |V| \\ 0 \leq k \leq p \\ I[j] = k}} x[1, j, k];$$

$$\varphi[2] = \bigwedge_{\substack{1 \leq j \leq |V| \\ 0 \leq k \leq p \\ F[j] = k}} x[N, j, k];$$

$$\varphi[3] = \bigwedge_{\substack{1 \leq i \leq N \\ 1 \leq j \leq |V| \\ 0 \leq k[1] < k[2] \leq p}} (\neg x[i, j, k[1]] \vee \neg x[i, j, k[2]]);$$

$$\varphi[4] = \bigwedge_{\substack{1\le i\le N\\ 1\le j\le |V|}} \bigvee_{0\le k\le p} x[i,j,k];$$

$$\varphi[5] = \bigwedge_{\substack{1\le i\le N-1\\ 1\le j[1]<j[2]\le |V|}} (\neg s[i,j[1]] \vee \neg s[i,j[2]]);$$

$$\varphi[6] = \bigwedge_{1\le i\le N-1} \bigvee_{1\le j\le |V|} s[i,j];$$

$$\varphi[7] = \bigwedge_{\substack{1\le i\le N-1\\ 1\le j[1]<j[2]\le |V|}} (\neg t[i,j[1]] \vee \neg t[i,j[2]]);$$

$$\varphi[8] = \bigwedge_{1\le i\le N-1} \bigvee_{1\le j\le |V|} t[i,j];$$

$$\varphi[9] = \bigwedge_{\substack{1\le i\le N-1\\ 1\le j\le |V|}} (\neg s[i,j] \vee \neg x[i,j,0]);$$

$$\varphi[10] = \bigwedge_{\substack{1\le i\le N-1\\ 1\le j\le |V|}} (\neg s[i,j] \vee x[i+1,j,0]);$$

$$\varphi[11] = \bigwedge_{\substack{1\le i\le N-1\\ 1\le j\le |V|}} (\neg t[i,j] \vee x[i,j,0]);$$

$$\varphi[12] = \bigwedge_{\substack{1\le i\le N-1\\ 1\le j\le |V|}} (\neg t[i,j] \vee \neg x[i+1,j,0]);$$

$$\varphi[13] = \bigwedge_{\substack{1\le i\le N-1\\ 1\le j[1]\le |V|\\ 1\le j[2]\le |V|}} (\neg s[i,j[1]] \vee \neg t[i,j[2]]);$$

$$\varphi[14] = \bigwedge_{\substack{1\le k\le p\\ 1\le i[1]<i[2]\le N-1}} (\neg y[k,i[1]] \vee \neg y[k,i[2]]);$$

$$\varphi[15] = \bigwedge_{1\le k\le p} \bigvee_{1\le i\le N-1} y[k,i];$$

$$\varphi[16] = \bigwedge_{\substack{1\le k\le p\\ 1\le i\le N-1\\ 1\le j\le |V|}} (\neg s[i,j] \vee \neg y[k,i] \vee x[i,j,k]);$$

$$\varphi[17] = \bigwedge_{\substack{1\le k\le p\\ 1\le i\le N-1\\ 1\le j\le |V|}} (\neg t[i,j] \vee \neg y[k,i] \vee x[i+1,j,k]);$$

$$\eta = \bigwedge_{r=1}^{17} \varphi[r].$$

Using standard transformations, we can obtain an explicit transformation η into ζ such that

$$\eta \Leftrightarrow \zeta$$

and ζ is a 3-CNF. Clearly, ζ gives us an explicit reduction from PPCR to 3SAT.

To obtain optimal solutions of PPCR we use genetic algorithms OA[1] (see [4]), OA[2] (see [5]), and OA[3] (see [6]) for the satisfiability problem. We have used heterogeneous cluster. Each test was runned on a cluster of at least 100 nodes. Selected experimental results are given in Table 1.

time	average	max	best
OA[1]	17.52 min	1.97 hr	3.12 min
OA[2]	15.89 min	2.45 hr	1.43 min
OA[3]	11.54 min	27.34 min	29 sec

Table 1: Experimental results for different genetic algorithms.

ACKNOWLEDGEMENTS. This research was partially financially supported by RFBR, research project No. 13-01-00048 a, and under the Agreement 02.A03.21.0006 of 27.08.2013 between the Ministry of Education and Science of the Russian Federation and Ural Federal University.

References

[1] P. Bendotti, P. Fouilhoux, and S. Kedad-Sidhoum, Permutation Problem using a unit-Capacity Robot for the nuclear Fuel Renewal in a Pressurized Water Reactor, *Septièmes Journées Franco-Chiliennes d'Optimisation, 2011.*

[2] P. Bendotti, P. Fouilhoux, and S. Kedad-Sidhoum, Permutation using a Unit-Capacity Robot, 2014. http://www-desir.lip6.fr/∼fouilhoux/documents/PPCR-Complexity.pdf

[3] M. Savelsbergh and M. Sol, The General Pickup and Delivery Problem, *Transportation Science*, **29** (1995) 17-29.
http://dx.doi.org/10.1287/trsc.29.1.17

[4] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Finding Regularities, *Applied Mathematical Sciences*, **6** (2012), 6577-6581.

[5] A. Gorbenko and V. Popov, The set of parameterized k-covers problem, *Theoretical Computer Science*, **423** (2012), 19-24.
http://dx.doi.org/10.1016/j.tcs.2011.12.052

[6] A. Gorbenko and V. Popov, Task-resource Scheduling Problem, *International Journal of Automation and Computing*, **9** (2012), 429-441.
http://dx.doi.org/10.1007/s11633-012-0664-y

Received: June 7, 2015; Published: September 9, 2015