Bounds and approximation results for scheduling coupled-tasks with compatibility constraints

R. Giroudeau[1], J.C König[1], B. Darties[2] and G. Simonin[3]

[1] LIRMM UMR 5506, 161 rue Ada 34392, Montpellier France
{rgirou,konig}@lirmm.fr
[2] LE2I UMR6306, Univ. Bourgogne Franche-Comté, F-21000 Dijon, France
benoit.darties@u-bourgogne.fr
[3] Insight Centre for Data Analytics, University College Cork, Ireland
gilles.simonin@insight-centre.org

Abstract. This article is devoted to propose some lower and upper bounds for the coupled-tasks scheduling problem in presence of compatibility constraints according to classical complexity hypothesis ($\mathcal{P} \neq \mathcal{NP}$, $\mathcal{ETH}$). Moreover, we develop an efficient polynomial-time approximation algorithm for the specific case for which the topology describing the compatibility constraints is a quasi split-graph.
Keywords: coupled-task, compatibility graph, complexity, approximation.

1 Introduction, motivations, model

We consider in this paper the coupled-task scheduling problem subject to compatibility constraints. The motivation of this model is related to data acquisition processes using radar sensors: a sensor emits a radio pulse (first sub-task a_i), and listen for an echo reply (second sub-task b_i). To make the notation less cluttered, the processing time of a sub-task will be denoted by a_i instead of p_{a_i} used in the theory of scheduling. Between these two instants (emission and reception), clearly there is an idle time L_i due to the propagation, in both sides, of the radio pulse. A coupled-task (a_i, L_i, b_i), introduced by Shapiro (1980), is a natural way to model such data acquisition. This model has been widely studied in several works, `i.e.` Błażewicz et al. (2009). Other works proposed a generalization of this model by including compatibility constraints: scheduling a sub-task during the idle time of another requires that both tasks are compatible. The relations of compatibility are modeled by a compatibility graph G, linking pair of compatible tasks only. This model is detailed in Simonin et al. (2012). In previous works, we studied the complexity of scheduling coupled-tasks with compatible constraints under several parameters like the size of the sub-tasks or the class of the compatibility graph (Simonin et al. 2013).

In this work, we propose original complexity and approximation results for the problem of scheduling *stretched* coupled-task with compatibility constraints. A *stretched* coupled-tasks i is a coupled-task having both sub-tasks processing time and idle time equal to a triplet $(\alpha(i), \alpha(i), \alpha(i))$, where $\alpha(i)$ is the *stretch factor* of the task i - one can apply a stretch factor $\alpha(i)$ to a reference task $(1, 1, 1)$ to obtain i -.

The objective is to minimize the makespan C_{max}. The input of the problem is a collection of coupled-tasks $\mathcal{T} = \{t_1, t_2, \dots t_n\}$, a stretch factor function $\alpha : \mathcal{T} \rightarrow \mathbb{N}$, and a compatibility graph $G_c = (\mathcal{T}, E)$ where edge from E link pairs of compatible tasks only. When dealing with stretched coupled-tasks only, a edge $\{x, y\} \in E$ exists if $\alpha(x) = \alpha(y)$ (then x and y can be scheduled together without idle time as the idle time of one task is employed to schedule the sub-task of the other, thus we can schedule sequentially a_x, a_y, b_x, b_y

- or a_y, a_x, b_y, b_x - in $\frac{4\alpha(x)}{3}$ time units), or if $3\alpha(x) \leq \alpha(y)$ (then x can be entirely executed during the idle time of y *i.e.* a_y, a_x, b_x, b_y and scheduling both tasks requires $3\alpha(y)$ time units). We note $\#(X)$ the number of different stretch factors in a set of tasks X, and we note $d_G(X)$ the maximum degree of any vertex $x \in X$ in a graph G_c.

We use the well-known Graham notation (Graham et al. 1979) to define the problems presented in this paper. In this work, we propose new complexity and inapproximability results when the compatibility graph is a restricted $1-$*stage bipartite* graph $G = (X, Y, E)$, `i.e.` a bipartite graph where edges are oriented from X to Y only. Then we show the problem is $\mathcal{NP}$-complete on a quasi-split graph $G = (G_X, G_Y, E)$[4] even if $\#(V(G_X)) = 1$ and $\#(V(G_Y)) = 1$, but is 5/4-approximable.

2 Complexity and approximation results

Theorem 1. *Deciding whether an instance of* $1|\alpha, G_c = 1 - stage - bipartite,\ \#(X) = 2, \#(Y) = 1, d_{G_c}(X) \in \{1,2\}, d_{G_c}(Y) \in \{3,4\}|C_{max}$ *is a problem hard to approximate within* $\frac{21-\rho^{\text{Max-3DM-2}}}{20} \leq \rho$, *where* $\rho^{\text{Max-3DM}}$ *gives the upper bound for the* Max-3DM. *Since* $\rho^{\text{Max-3DM-2}} \leq \frac{140}{141}$, *we obtain* $1 + \frac{1}{2820}$.

Proof. We prove first that the problem is $\mathcal{NP}$-complete via a polynomial-time reduction. Based on this reduction, we apply the gap-preserving reduction.

The proof is based on a reduction from the maximum 3 Dimensional Matching (Max-3DM) (Garey & Johnson 1979): let A, B, and C be three disjoint sets of equal size, with $n = |A| = |B| = |C|$, and a set $T \subseteq A \times B \times C$ of triplet, with $|T| = m$. The aim is to find a matching (set of mutually disjoint triplets) $T^* \subseteq T$ of maximum size. This problem is well known to be $\mathcal{NP}$-complete. The restricted version of this problem in which each element of $A \cup B \cup C$ appears exactly twice is denoted Max-3DM-2 and remains $\mathcal{NP}$-complete (Chlebik 2003). In this restricted version, we have $m = 2n$.

We transform the instance of Max-3DM-2 to an instance of $1|\alpha, G_c = 1 - stage\ bi partite,\ \#(X) = 2, \#(Y) = 1, d_{G_c}(X) \in \{1,2\}, d_{G_c}(Y) \in \{3,4\}|C_{max} = 63n - 3k(1-\epsilon)$ as follows: we define a set of tasks $X \cup Y$ and model the compatibility constraint with a graph $G_c = (X, Y, E)$. For each element $x_i \in A \cup B \cup C$, we add an *item* coupled-task x_i into X with $\alpha(x_i) = 1$. For each triplet $t_i \in T$, we add a *box* coupled-task t_i to Y with $\alpha(t_i) = 9$, and an *item* coupled-task t'_i with $\alpha(t'_i) = 2 + \epsilon$. For each $t_i \in T$ and each $x_i \in t_i$, we add the compatibility arc (x_i, t_i) to E. We also add the compatibility arc (t'_i, t_i) to E. So, the set of X-tasks (resp. Y-tasks) are constituted by *item* coupled-task x_i and t'_i (resp. *box* coupled-task).

Clearly we have m *box* coupled-tasks (each with an idle time of 9 units) of degree 4 in G_c, m *item* coupled-tasks with stretch factor $2 + \epsilon$ of degree 1 in G_c, and $3n$ *item* coupled-tasks with stretch factor 1 of degree 2 in G_c. Moreover G_c is a bipartite graph. The reduction is constructed in polynomial time.

It exists a schedule of length $63n - 3k(1-\epsilon)$ iff it exists a matching of size k for Max-3DM-2 instance.

Hereafter, we propose some negative results concerning the existence of subexponential-time algorithms under the following complexity-theoretic hypothesis that is known as the Exponential-Time Hypothesis (see (Woeginger 2001) for a survey on exact algorithms for $\mathcal{NP}$-hard problems) for stretched coupled-tasks, and other ones previously studied.

Recall first the Exponential-Time Hypothesis ((Impagliazzo & Paturi 2001), and (Impagliazzo et al. 2001)): there exists a constant $c > 1$ such that there exists no algorithm for $3-$Satisfiability that uses only $O(c^l)$ time where l denotes the number of variables.

[4] A quasi split graph is a connected graph $G = (G_X, G_Y, E)$, with G_X a connected non-oriented graph (not complete) and G_Y a independent set. The other arcs are oriented from X to Y only.

Corollary 1. *Assuming the Exponential-Time Hypothesis, there exists no algorithm with a worst-case running time that is subexponential in n (the number of vertices), i.e.:*

1. *For the* $1|a_i = b_i = p, L_i = 2p, G_c|C_{max}$ *problem in* $O(2^{o(n)})$ *time*
2. *For* $1|a_i = a, b_i = b, L_i = a + b, G_c|C_{max}$ *in* $O(2^{o(n)})$ *time*
3. $1|\alpha, G_c = 1 - bipartite|C_{max}$ *in* $O(2^{O(n)})$*-time algorithm.*

Proof. 1. For $1|a_i = b_i = p, L_i = 2p, G_c|C_{max}$: In (van Rooij et al. 2013), the authors proved that for Partition into triangles on graphs of maximum degree four, there is no algorithm with a worst-case running time $O(2^{o(n)})$ that is subexponential in n. Therefore, we transform a Partition into triangles instance with n vertices and m edges into an equivalent instance G_c for bounded degree at most four. Since the transformation is linear (see (Simonin et al. 2012)) the result holds.
2. For the problem $1|a_i = a, b_i = b, L_i = a+b, G_c|C_{max}$: In (Lokshtanov, Marx & Saurabh 2011) the authors proved that for Hamiltonian path there is no $O(2^{o(n)})$-time algorithm. As the same way as previously the transformation is linear (see (Simonin et al. 2012)).
3. $1|\alpha, G_c = 1-bipartite|C_{max}$: In (Chen, Jansen & Zhang 2014), the authors proved that for Max 3DM, there is no $O(2^{O(n)})$-time algorithm, therefore this result is transposed to the scheduling problem using the first part of the proof of Theorem 1.

Theorem 2. *Scheduling stretched coupled task in presence of a quasi split graph is a* $\mathcal{NP}-$*complete problem even if* $\#(V(G_X)) = 1$ *and* $\#(V(G_Y)) = 1$

Proof. The proof is based on a reduction from a variant of the well-know $\mathcal{NP}$-complete Partition into triangles. This problem consists to ask if the vertices of a graph $G = (V, E)$, with $|V| = 3q, q \in \mathbb{N}$, can be partitioned into q disjoints sets $T_1, T_2, \ldots, T_q$, each containing exactly three vertices, such that for each $T_i = \{u_i, v_i, w_i\}, 1 \leq i \leq q$, all three of the edges $\{u_i, v_i\}, \{u_i, w_i\}, \{w_i, v_i\}$ belong to E.

The problem Partition into triangles remains $\mathcal{NP}$-complete even if the graph G can be partitioned into three sets with the same size, A, B et C such that each set is an independent set (Morandini, M. 2004). The polynomial-time transformation is based on this variant. Let $G = (A \cup B \cup C, E)$ be an instance of the variant of Partition into Triangles. We consider the split-graph $G' = (A \cup B, C, E')$ obtained as follows:

$\forall v \in A$ (resp. B), we create a vertex A_v (resp. B_v) with processing time $(1, 1, 1)$. Moreover, $\forall v \in C$ we create a task C_v with processing time $(4, 4, 4)$. The edges between A and B remained the same as the G' whereas the edge between $A \cup B$ and C are oriented. Finally in order to have a connected graph, we add two news vertices (resp. one) z_0 and z_1 (resp. z_2 with processing time equal to $(1, 1, 1)$ (resp. $(4, 4, 4)$). We add edges between z_0 to A_v (resp. z_1 to B_v). Lastly, we add the three edges (z_0, z_2), (z_1, z_2) and (z_0, z_1).

Notice that the graph $A_v \cup B_v$ form a bipartite graph. The problem is clearly in $\mathcal{NP}$. It exists a positive solution for the variant of Partition into triangles iff a valid schedule of length $12 \times (|C| + 1)$ exists. It is sufficient to execute the two tasks A_v and $B_{v'}$ in four units of time into a task C_u.

Theorem 3. *The problem is* $5/4$*-approximable on quasi split-graph where* $\#(V(G_Y)) = 1$*.*

Proof. W.l.o.g., we suppose that the processing time of X-tasks (resp. Y-tasks) is $(1, 1, 1)$ (resp. $\alpha(y_i)$). Indeed, if $\alpha(x) > 1$, we put $\alpha(y_i) = \lfloor \frac{\alpha(y_i)}{\alpha(x)} \rfloor$ and $\alpha(x) = 1$.

Algorithm: we transform the problem into an oriented maximum flow-problem between G_X and G_Y with two sources s and t, with $\omega(s, x) = \omega(x, y) = 1$ and $\omega(y, t) = \lfloor \frac{\alpha(y_i)}{3\alpha(x)} \rfloor, \forall y_i \in Y, \forall x \in X G_Y$ where $\omega(i, j)$ is the capacity of an arc (i, j) . After the computation of a maximum flow F of value f, for the uncovered remaining X-tasks a maximum

M-matching ($|M| = m$) is applied. The schedule consists in processing first, the Y-tasks with X-tasks inside. The M-tasks are executed after. Lastly, we schedule s isolated-tasks. The length of schedule given by the algorithm is $C_{max} \leq \sum_{y_i \in Y} 3\alpha(y_i) + 4m + 3s$ with $2m + s + f = n = |X|$ and $\sum_{y_i \in Y} 3\alpha(y_i) \geq 9f$. In similar way, the optimal length is $C^*_{max} \geq \sum_{y_i \in Y} \alpha(y_i) + 4m^* + 3s^*$. We suppose that in Y-tasks where are p^*-edges processed and r^* isolated-tasks, then we obtain $2(p^* + m^*) + r^* + s^* = n$, $p^* + r^* \leq f$, and $\sum_{y_i \in Y} \alpha(y_i) \geq 12p^* + 9r^*$. In the worst-case, the p^*-edges are split into two tasks (so p^* news tasks are added to s^*), and also the matched-edges are split (for each m^* edges one task is executed into the Y-task, instead of one of r^*-tasks). Therefore, $2m^*$ tasks are added to the s-value. In the worst case, we have $m^* = r^*$, $s = s^* + p^* + 2r^*$ and $m = 0$. In such case, $C_{max} \leq 12p^* + 9r^* + 3s^* + 3p^* + 6r^*$ and $C^*_{max} = 12p^* + 9r^* + 4r^* + 3s^*$. Thus $\rho \leq \frac{15p^*+15r^*+3s^*}{12p^*+13r^*+3s^*} \leq \max(5/4, 15/13, 1) = 5/4$.

Tightness: it exists an example for the $C^*_{max} = 36$, and for the heuristic $C_{max} = 45$. Consider the graph: three triangles (x_1, x_2, y_1), (x_3, x_4, y_2), and (x_5, x_6, y_1). We add the edges $(x_2, y_3$, (x_3, y_1) and (x_5, y_2). The optimal solution consists in executing the X-tasks into the Y-tasks; whereas the heuristic leads the solution in which three X-tasks are processed after the Y-tasks.

References

Blażewicz, J., Ecker, K., Kis, T., Potts, C., Tanas, M. & Whitehead, J. (2009), Scheduling of coupled tasks with unit processing times, *Technical report, Poznan University of Technology.*

Chen L., Jansen K. & Zhang G. (2014), On the optimality of approximation schemes for the classical scheduling problem, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*

Chlebík M. & Chlebíková J. (1979), Inapproximability results for bounded variants of optimization problems, *Electronic Colloquium on Computational Complexity* **10** (26), pp. 1–26.

Garey, M. R. & Johnson, D. S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA.

Graham, R. L., Lawler, E. L., Lenstra, J. K. & Kan, A. H. G. Rinnooy (1979), Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey, *Annals of Discrete Mathematics* **5**, pp. 287–326.

Impagliazzo, R. & Paturi, R. (2001), On the Complexity of k-SAT, *Journal of Computer and System Sciences* **62**(2), 367–375.

Impagliazzo, R., Paturi, R. & Zane, F. (2001), Which Problems Have Strongly Exponential Complexity?, *Journal of Computer and System Sciences* **63**(4), 512–530.

Lokshtanov, D., Marx, D. & Saurabh, S. (2011), Lower bounds based on the Exponential Time Hypothesis, *Bulletin of the EATCS* **105**, 41–72.

Morandini, M. (2004), NP-complete problem: partition into triangles, *Technical report, Universita' di Udini.*

Shapiro, R. D. (1980), Scheduling coupled tasks, *Naval Research Logistics Quarterly* **27**, 477–481.

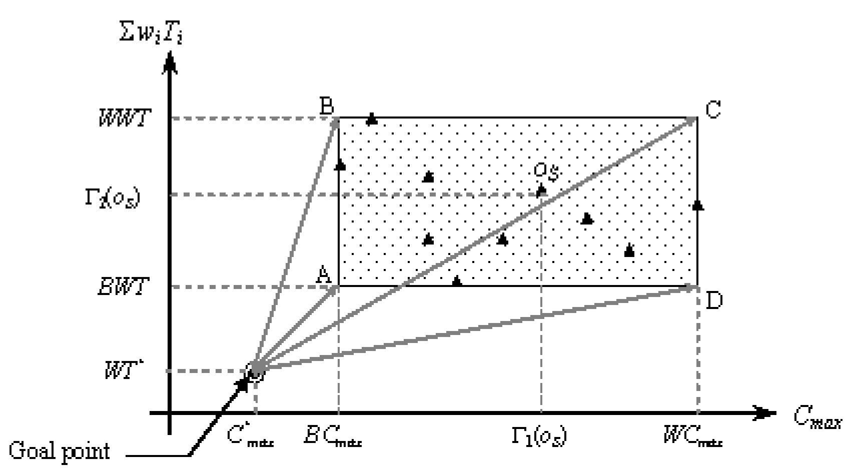
Simonin, G., Darties, B., Giroudeau, R. & König, J.-C. (2011), Isomorphic coupled-task scheduling problem with compatibility constraints on a single processor, *J. of Scheduling* **14**(5), 501–509.

Simonin, G., Giroudeau, R. & König, J.-C. (2013), Approximating a coupled-task scheduling problem in the presence of compatibility graph and additional tasks', *International Journal of Planning and Scheduling* **1**(4),pp. 285–300.

Simonin, G., Giroudeau, R., König, J.-C. & B. Darties (2012), Theoretical Aspects of Scheduling Coupled-Tasks in the Presence of Compatibility Graph', *Algorithmic in Operations Research* **7**(1), 1—-12.

van Rooij J. M. M., van Kooten Niekerk, M. E. & Bodlaender H. L. (2013), Partition Into Triangles on Bounded Degree Graphs, *Theory Computing Systems* **52**(4), pp. 687-718.

Woeginger, G. J. (2001), Exact Algorithms for NP-Hard Problems: A Survey, *in* Combinatorial Optimization, pp. 185–208.



$\Sigma w_i T_i$
WWT
$\Gamma_2(o_S)$
BWT
WT^*
B
C
o_S
A
D
C_{max}
C^*_{max}
BC_{max}
$\Gamma_1(o_S)$
WC_{max}
Goal point