Divide and Conquer in Multi-agent Planning

Eithan Ephrati
Computer Science Department
University of Pittsburgh
Pittsburgh, PA
tantush@cs.pitt.edu

Jeffrey S. Rosenschein
Institute of Computer Science
The Hebrew University
Jerusalem, Israel
jeff@cs.huji.ac.il

Abstract

In this paper, we suggest an approach to multi-agent planning that contains heuristic elements. Our method makes use of subgoals, and derived sub-plans, to construct a global plan. Agents solve their individual sub-plans, which are then merged into a global plan. The suggested approach may reduce overall planning time and derives a plan that approximates the optimal global plan that would have been derived by a central planner, given those original subgoals.

We consider two different scenarios. The first involves a group of agents with a common goal. The second considers how agents can interleave planning and execution when planning towards a common, though dynamic, goal.

Decomposition Reducing Complexity

The complexity of a planning process is measured by the time (and space) consumed. Let b be the branching factor of the planning problem (the average number of new states that can be generated from a given state by applying a single operator), and let d denote the depth of the problem (the optimal path from the initial state to the goal state). The time complexity of the planning problem is then $O(b^d)$ (Korf 1987).

In a multi-agent environment, where each agent can carry out each of the possible operators (possibly with differing costs), the complexity may be even worse. A centralized planner should consider assigning each operator to each one of the n agents. Thus, finding an optimal plan becomes $O(n \times b)^d$.

However, if the global goal can be decomposed into n subgoals ($\{g_1, \ldots, g_n\}$) the time complexity may be reduced significantly. Let b_i and d_i denote respectively the branching factor and depth of the optimal plan that achieves g_i. Then, as shown by Korf in (Korf 1987), if the subgoals are independent or serializable,[1] the central multi-agent planning time complexity can be reduced to $\sum_i((n \times b_i)^{d_i})$, where $b_i \approx \frac{b}{n}$ and $d_i \approx \frac{d}{n}$.

[1] A set of subgoals is said to be *independent* if the plans that achieve them do not interact. If the subgoals are *serializable* then there exists an ordering among them such that achieving any subgoal in the series does not violate any of its preceding subgoals.

This phenomenon of reduced complexity due to the division of the search space can be exploited most naturally in a multi-agent environment. The underlying idea is to assign to each agent a subgoal and let that agent construct the plan that achieves it. Since agents plan in parallel, planning time is further reduced to $\max_i(n \times b_i)^{d_i}$. Moreover, if each agent is to generate its plan according to its own view (assuming that the available operators are common knowledge) then the complexity becomes $\max_i(b_i)^{d_i}$. The global plan can then be constructed out of local plans that are based upon the agents' local knowledge. Unfortunately, unless the subgoals are independent or serial, the plans that achieve the set of subgoals interfere, and conflicts (or redundant actions) may arise and need to be resolved.

In this paper we suggest a heuristic approach to multi-agent planning that exploits this phenomenon of decomposed search space. The essential idea is that the individual sub-plans serve to derive a heuristic function that is used to guide the search for the *global plan*. This global search is then done in the space of world states which is pruned using the A^* algorithm. Our method makes use of pre-existing subgoals. These subgoals are not necessarily independent, nor are they necessarily serial. The separate agents' sub-plans, each derived separately and in parallel, are ultimately merged into a unified, valid global plan. The suggested approach may reduce overall planning time while deriving the optimal global plan that would have been derived, given those original subgoals. In multi-agent environments this approach also removes the need for a central planner that has global knowledge of the domain and of the agents involved.

Our scenario involves a group $\mathcal{A} = \{a_1, \ldots, a_n\}$ of n agents. These agents are to achieve a global goal G. The global goal, G, has been divided into n subgoals ($\{G_1, \ldots, G_n\}$), and formulated as a subgoal planning problem (i.e., the interrelationship among subgoals has been specified). The agents communicate as they construct a global plan.

A Simple Example

Consider a scenario in the slotted blocks world. As described in Figure 1 there are three agents (a_1, a_2, a_3)

and 4 blocks (a,b,c,d) with lengths of 1, 2, 2, and 3 respectively. The world may be described by the following relations: **Clear**(b)(—there is no object on b); **On**($b, x, V/H$)(—b is located on block/location x either vertically (V) or horizontally (H)); **At**(x, loc)(—the left edge of object x (agent or block) is at loc).

The functions $r(b)$ and $l(b)$ return the region of b's left edge, and the length of b, respectively. We will use only the first letter of a predicate to denote it.

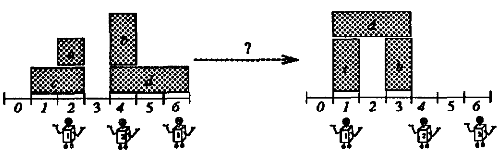

Figure 1: An Arch in the Blocks World

The available operators (described in a STRIPS-like fashion) are:

Take$_i(b, x, y)$— Agent i takes b from region/block x to region/bloc y: [`cost:` $|loc(x) - loc(y)| \times l(b)$, `pre:` $C(b)$,$C(y)$,$A(a_i, x)$, `del:` $C(y)$,$A(a_i, x)$,$A(b, x)$,$O(b, x, z)$, `add:` $C(x)$,$O(b, y, z)$,$A(a_i, y)$,$A(b, y)$]

Rotate$_i(b)$—i rotates b by $\pm\frac{\pi}{2}$: [`cost:` $l^2(b)$, `pre:` $C(b), A(a_i, r(b))$, `del:` $O(b, x, z)$, `add:` $O(b, x, \bar{z})$] ($\bar{H}$ denotes V and vice versa)

Move$_i(x, y)$—i goes from x to y: [`cost:` $|x - y|$, `pre:` $A(a_i, x)$, `del:` $A(a_i, x)$, `add:` $A(a_i, y)$]

The initial state is described in the left side of Figure 1. The agents are to construct an arch such as the one pictured in the right side of the figure. A straightforward division into subgoals is to first construct left and right columns (appropriately distant and aligned) and then put up a top. Given this *a priori* breakdown into subgoals, our agents are to go through a planning process that will result in satisfying the original goal.

Assumptions and Definitions

- The *global goal*, G, is a set of predicates, possibly including uninstantiated variables. g denotes any grounded instance of G (a set of grounded predicates that specifies a set of states). We assume that G is divided into n abstract subgoals $\{G_1, G_2, \ldots, G_n\}$, such that there exists a consistent set of instances of these subgoals that satisfies G ($\cup_i g_i \models G$).

  In accordance with the possibly required (partial) order of subgoal achievement, we denote the preconditions for any plan, p_i, that achieves g_i by g_i^0 (which for most subgoals is simply the initial state).
- Each p_i is expressed by the set of the essential propositions that enable any sequence of operators that construct it. These propositions are partially ordered according to their temporal order in p_i.[2]
- For the merging/composition process to find the (optimal) global plan, it will, in general, be necessary to generate more than just one plan for some subgoals. d_i denotes the depth (radius) of the search that is needed so as to generate the sufficient number of sub-plans that achieve g_i. We assume that d_i is known ahead of time (for each g_i).[3] $\mathcal{P}_i$ denotes the (sufficient) set of plans that is generated within this d_i-depth search.
- Each agent has a *cost function* over the domain's operators. The cost of a_j's plan $c_j(p_j)$ is $\sum_{k=1}^{m} c_j(op_k)$.
- Given that the set of propositions E holds (in some world state), $F^1_{ollow}(E)$ is defined to be the set of all propositions that can be satisfied by invoking at most one operator at that state. ($F^1_{ollow}(E) = \{I \mid \exists_{op}[op(E) \models I]\}$ where $op(E)$ denotes the invocation of op at a state that satisfies E.) Similarly, $F^2_{ollow}(E)$ is the set of propositions that can be achieved by invoking at most two operators *simultaneously* (by two agents) given E, and $F^n_{ollow}(E)$ is the set that can be achieved by at most n simultaneous actions.

The Process

At the beginning of the planning process each agent, i, is assigned (for the purposes of the *planning process*) one subgoal g_i. Given that subgoal, the agent derives $\mathcal{P}_i$, the (sufficient) set of sub-plans that achieves it given some initial configuration g_i^0.

The significant savings in time and space complexity of the search is established by the decomposition of the search space and by parallelism of the search. However, the primary phase of the subgoal technique is the process of merging the sub-plans that achieve the given subgoals. A sub-plan is constructed by an agent with only a local view of the overall problem. Therefore, conflicts may exist among agents' sub-plans, and redundant actions may also have been generated. Given the set of sub-plans, we are looking for a method to inexpensively merge them into an optimal global plan.

To do so we employ an iterative search. The underlying idea is the *dynamic generation of alternatives* that identifies the optimal global plan. At each step, all agents state additional information about the sub-plan of which they are in charge. The current set of candidate global plans is then expanded to comprise the new set of candidate global plans. The process continues

[2] Using SNLP (McAllester & Rosenblitt 1991) terminology, these are the propositions in the causal links that construct the "nonlinear abstraction" of p_i, partially ordered according to their safety conditions, and stated as prerequisites (preconditions in UCPOP's terminology (Penberthy & Weld 1992)) (e.g., if step w has prerequisite $On(x, B)$ and step s enables it by establishing $On(A, B)$, the essential proposition is $On(x, B)$ rather than $On(A, B)$).

[3] This unrealistic assumption is needed only for the *completeness* of the planning process. However, using domain dependent knowledge, the corresponding d_i's may be assessed heuristically. In general, the more the sub-plans will tend to interact (and the closer to optimal the solution needs to be) the deeper the d_i's that are needed.

until the optimal plan is found. Plans are represented by the partially ordered sets of the essential propositions that enable them. These sets of propositions are aggregated throughout the process.

We use ordered propositions instead of sequences of operators for the following reasons. First, the constructed sub-plans serve only to *guide* the heuristic search for the actual global multi-agent plan. The *actual* multi-agent action to establish a proposition is determined only during the merging process itself. This is essential for the efficiency of the resulting multi-agent plan.[4] Second, the propositions encode all the information needed for the heuristic evaluation. And third, by dealing with propositions we achieve more flexibility (least commitment) in the merging process, both in the choice of operators and in their bindings.

Note that the essential search method is similar to the search employed by progressive world-state planners. In general, the search through the space of states is inferior to the search, as conducted by POCL planners, through the space of plans (Minton, Bresina, & Drummond 1991). The reason is that any (nondeterministic) choice of action within the first method also enforces the timing of that action (and thus, a greater breadth of search is needed to ensure completeness, e.g., in the Sussman anomaly). However, given the individual sub-plans, our merging procedure need consider only a small number of optional expansions, among which the heuristic evaluation "foresees" most commitments that may result in backtracking. Thus, becomes possible and worthwhile to avoid the causal-link-protection step of the POCL planners.

To achieve that, the search method employs an A^* algorithm where each path represents one optional global multi-agent plan. The heuristic function ($f' = g+h'$) that guides the search is dynamically determined by the agents during the process. g is the actual cost of the partial path (multi-agent plan) that has already been constructed. h' is the sum of the approximate remaining costs, h'_i, that each agent assigns to that path, based on its own generated sub-plan. Since based upon an actually constructed plan, each individual estimate, h'_i, is absolutely accurate in isolation. Thus, if the subgoals are independent, then the global heuristic function ($\sum_i^n h'_i$) will be accurate, and the merging process will choose the correct (optimal) candidate for further expansion at each step of the process.

Unfortunately, since in general sub-plans will tend to interfere with one another, h' is an underestimate (the individual estimates will turn out to be too optimistic). An underestimated heuristic evaluation is also desirable, since it will make the entire A^* search admissible, meaning that once a path to the global goal has been found, it is guaranteed to be the optimal one. However, due to overlapping constraints ("favor relations" (Martial 1990), or "positive" interactions) the global heuristic evaluation might sometimes be an overestimate. Therefore, the A^* search for the optimal path would (in those cases) have to continue until the potential effect of misjudgment in the global heuristic evaluation fades away. In general, the more overlap appears in the individual sub-plans, the more additional search steps are needed.

More specifically, the agents go through the following search loop:[5]

1. At step k one aggregated set (of propositions), A_j^{k+}, is chosen from all sets with minimal heuristic value, $\mathcal{A}^{k+}$. This set (with its corresponding multi-agent plan) is the path currently being considered. Each agent declares the maximal set of propositions, E_i^*, such that:

   (a). These propositions represent some possible sequence of consecutive operators in the agents' private sub-plan, and all their necessary predecessors hold at the current node.

   (b). The declaration is "feasible," i.e., it can be achieved by having each of the agents perform at most one action simultaneously with one another ($E_i^* \subseteq F_{ollow}^n(A_j^{k+})$).

2. All (set-theoretic) maximal feasible expansions of A_j^{k+} with elements of the agents' declarations are generated. [Each expansion, $Ex(A_j^{k+})$, is one of the fixed points $\{I \mid (I \in \bigcup_i E_i^*) \wedge (I \cup Ex(A_j^{k+}) \in F_{ollow}^n(A_j^{k+})\}$. Note that this is only a subset of the expansions that a "blind planner" should generate.]

3. At this stage, based on the extended set of propositions, the agents construct additions to the ongoing candidate plans. Each expansion that was generated in the previous step induces a sequence of operations that achieves it. The generation of these sequences is discussed below.

4. All expansions are evaluated, in a central manner, so as to direct the search (i.e., find the value, $f' = g+h'$, of the A^* evaluation function):

   (a). The g component of each expansion is simply taken to be the cost of deriving it (the cost of the plan that is induced by the current path plus the additional cost of the multi-agent plan that derives the expansion).

   (b). To determine the heuristic component, h', each agent declares h'_i, the estimate it associates with

[4]For example, it might be the case that towards the achievement of his assigned subgoal a_i planned to perform op_k in order to establish proposition P, but in the multi-agent plan P will actually be established by a_j performing op_r. Therefore, what counts for the global plan is *what* is established, rather than *how* it is established.

[5]The set of all aggregated sets of propositions at step k is denoted by $\mathcal{A}^k$ (its constituent sets will be denoted by A_j^k, where j is simply an index over those sets). $\mathcal{A}^{k+}$ denotes the set that has the maximal value according to the heuristic function at step k.

each newly-formed set of aggregated propositions. This is the cost it associates with completing its "private" sub-plan, given that the expansion is established. The h' value is then taken to be the sum of these estimates $(\sum_i h'_i(Ex_r(A^{k+})))$.

5. The aggregated set A_j^{k+} is replaced by its union with all of its expansions: $\mathcal{A}^{k+1} = (\mathcal{A}^k \setminus A_j^{k+}) \cup \{A_j^{k+} \cup Ex_r(A_j^{k+})\}$.

The process ends when all "best plans" have been found. Since the heuristic function is not guaranteed to be an underestimate, stopping when the first global plan has been generated may not result in the optimal global plan. It is a matter of policy, how much more searching the agents should do (if at all) to discover better global plans.

The entire process has the following advantages from a complexity point of view. First, the branching factor of the search space is strictly constrained by the individual plans' propositions. Second, the A^* algorithm uses a relatively good heuristic function, because it is derived "bottom-up" from the plans that the agents have already generated (not simply an artificial h' function). Third, generation of successors in the search tree is split up among the agents (each doing a part of the search for a successor). Fourth, the heuristic function is calculated only for maximally "feasible" alternatives (infeasible alternatives need not be considered).

Theorem 1 *Given $\mathcal{P}_i$, the sub-plans that achieve each subgoal $\{g_i\}$, the merging algorithm will find the optimal multi-agent plan that achieves these subgoals. The process will end within $O(q \times d)$ steps where d is the length of the optimal plan, and q is a measure of the positive interactions between overlapping propositions.*

In comparison to planning by a central planner, the overall complexity of the planning process, $O((n \times b)^d)$, is reduced to $O(\max_i b_i^{d_i} + b \times n \times q \times d)$, where $b_i^{d_i} \approx (\frac{b}{n})^{\frac{d}{n}}$.

Proof: The formal proofs of the theorems in this paper appear in (Ephrati 1993).

Construction of the Global Plan

The multi-agent plan is constructed throughout the process (Step 3 of the algorithm). At this step, all the optimal sequences of operators are determined. We require that the actual plan be constructed dynamically in order to determine the g value of each alternative. The construction of the new segments of plans is determined by the cost that agents assign to each of the required actions; each agent bids for each action that each expansion implies. The bid that each agent gives takes into consideration the actions that the agent has been assigned so far. Thus, the global minimal cost sequence can be determined.

An important aspect of the process is that each expansion of the set of propositions belongs to the F^n_{ollow} of the already achieved set. Therefore, it is straightforward to detect actions that can be performed in parallel. Thus the plan that is constructed is not just cost-efficient, but also time-efficient.

There are several important tradeoffs to be made here in the algorithm, and the decision of the system designer will affect the optimality of the resulting plan. First, it would be possible to use Best First Search instead of A^* so as to first determine the entire set of propositions, and only then construct the induced plan. Employing such a technique would still be less time-consuming than global planning. Second, when the agents add on the next steps of the global plan, they could consider the (developing) global plan from its beginning to the current point when deciding on the least expensive sequence of additional steps. This will (eventually) result in a globally optimal plan, but at the cost of continually reevaluating the developing plan along the way. Alternatively, it is possible to save all possible combinations of the actions that achieve any $Ex_r(A_j^{k+})$, and thus have a set of plans correspond to each expansion. Third, each agent declares $E_i^* \subseteq F^n_{ollow}(\mathcal{A}^{k+})$ to ensure maximal parallelism in the resulting global plan. However, agents may relate just to $F^1_{ollow}(\mathcal{A}^{k+})$ and establish parallelism only after the global plan is fully constructed.

Back to the Example

Consider again the example. Assume that the agents' subgoals are (respectively): $g_1 = \{A(c,1), O(c,1,V), C(c)\}$, $g_2 = \{A(b,3), O(b,3,V), C(b)\}$, and $g_3 = \{A(d,1), O(d,c,V)\}$. To simplify things we will use throughout this example $F^1_{ollow}(\mathcal{A}^{k+})$ instead of $F^n_{ollow}(\mathcal{A}^{k+})$. The resulting multi-agent plan is illustrated in Figure 2.

Given these subgoals, the agents will generate the following sets of propositions:[6]

$p_1 = \{[C(a), \underline{A(c,1)}, A(a_i, r(a))]^{[0]} \cup [\underline{C(c)}]^{[1]} \cup [A(a_i, r(c))]^{[2]} \cup [\underline{O(c,1,V)}]^{[4]}\}$ (this ordered set corresponds to the plan $\langle T_1(a,2,3), M_1(3,1), R_1(c)\rangle$).

$p_2 = \{[\underline{C(b)}, A(a_j, r(b)), C(3)]^{[0]} \cup [\underline{O(b,3,V)}]^{[2]}\}$ (inducing the plan $\langle T_2(b,4,3)\rangle$).

$p_3 = \{[C(b), C(3)]^{[0]} \cup [A(a_k, r(b)), C(3)]^{[2]} \cup [\underline{C(d)}]^{[2]} \cup [A(a_k, r(d))]^{[1]} \cup [\underline{g_2}, \underline{g_1}, \underline{O(d,c,H)}]^{[9]}\}$ (inducing the plan $\langle M_3(6,4), T_3(b,4,3), M_3(3,4), T_3(d,4,1)\rangle$).

Notice that there exists a positive relation between a_2's and a_3's sets of propositions (both would need block b to be removed from on top of block d), but there is a possible conflict, slot 3, between their plans and a_1's plan.

At the first iteration, there is only one candidate set for expansion—the empty set. The aggregated set

[6]We underline propositions that, once satisfied, must stay valid throughout the process (e.g., propositions that construct the final subgoal). The region $\bar{b}$ denotes any region besides $r(b)$. We use only the first letter of operators and predicates to denote them. The additional estimated cost of satisfying a subset of propositions appears in the superscript brackets.

of declared propositions is: $[\underline{A(c,1)}, C(a), \underline{C(b)}, C(3), A(a_i, r(a)), A(a_j, r(b))]$. The (sole) expansion is fully satisfied by the initial state; therefore, $g(A^1) = 0$, and $f'(A^1)$ is equal to its h' value (that is, the sum of the individual estimate costs, which is 23, i.e., $= 7 + 2 + 14$, a heuristic overestimate of 4).
At the second iteration, a_1 declares $[\underline{C(c)}]$, a_2 declares $[\underline{O(b,3,V)}]$, and a_3 may already declare $[\underline{C(d)}]$. All declarations are in $F^1_{ollow}(A^1)$. Thus, $Ex(A^1) = [\underline{C(c)}, \underline{O(b,3,V)}, \underline{C(d)}])$. These propositions can be achieved by $T_i(a,2,0)$ and $T_j(b,4,3)$. The bids that a_1,a_2 and a_3 give to these actions are respectively [2,5], [4,2], and [6,4]. Therefore, a_1 is "assigned" to block a and a_2 is assigned to block b while a_3 remains unemployed. The constructed plan is $\langle\{T_1(a,2,0), T_2(b,4,3)\}\rangle$ (where both agents perform in parallel), yielding a g value of 4.
At the third iteration, $(A^{2+} = [\underline{C(c)}, \underline{O(b,3,V)}, \underline{C(d)}])$, a_1 declares $[A(a_i, r(c))]$ and a_3 declares $[A(a_k, \underline{r(d)}), C(c)]$. According to the agents' bids, this expansion can best be achieved by $\langle\{M_1(0,1), M_2(3,4)\}\rangle$
At the fourth iteration, only a_1 has a feasible expansion to the current best set, that is $[O(c,1,V)]$ (note that a_3 may not declare his final subgoal before a_2's and a_1's assigned subgoals are satisfied). The corresponding segment of the multi-agent plan is $\langle R_1(c)\rangle$.
Finally, **at the fifth iteration,** only a_3's assigned goal is not satisfied, and he declares $[\underline{O(d,c,H)}]$. This last expansion is best satisfied by $\langle \underline{T_2(d,4,1)} \rangle$. Thus, the overall cost is 19. Notice that the final goal is achieved without any physical contribution on the part of a_3.

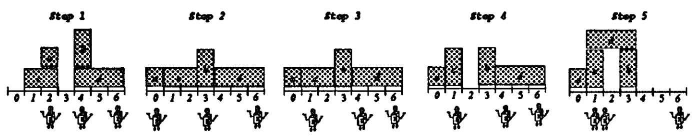

Figure 2: The resulting multi-agent plan

Interleaved Planning and Execution

The multi-agent planning procedure is based on the *incremental* process of merging sub-plans. This attribute of the process makes it very suitable for scenarios where the execution of the actual plan is urgent. In such scenarios it is important that, parallel to the planning process, the agents will actually execute segments of the plan that has been constructed so far (Dean & Boddy 1988; Durfee 1990). We assume that there is some look-ahead factor, l, that specifies the number of planning steps that should precede the actual execution step(s). We also assume that each agent can construct the first l optimal steps (in terms of propositions) of its own sub-plan.

The fact that in order to find the first step(s) of the multi-agent optimal plan it is important for the merging process to have only the corresponding first step(s) of the individual sub-plans, also makes the process very suitable for scenarios where the global goal may change dynamically. In such cases, the required revision of the (merged) multi-agent plan may sufficiently be expressed only by the first l look-ahead steps. Moreover, the process is flexible in response to such global changes, since they may be handled through the division of the new goal into subgoals. Thus, a change in the global goal may be reflected only in changes in several subgoals, and plan revision is needed only in several sub-plans.

We can therefore use the planning algorithm in scenarios where planning and execution are interleaved, and goals may dynamically change. As in the previous scenario, the key element of the approach is a cost-driven merging process that results in a coherent global plan (of which the first l sets of simultaneous operators are most relevant), given the sub-plans. At each time step t each agent, i, is assigned (for the purposes of the *planning process*) one task and derives (the first l steps of) p_i^t, the sub-plan that achieves it. Note that once i has been assigned g_i^t at any given t, the plan it derives to accomplish the subgoal stays valid (for the use of the algorithm) as long as g_i^t remains the same. That is, for any time $t+k$ such that $g_i^{t+k} = g_i^t$, it holds that $p_i^{t+k} = p_i^t$. Thus, re-planning is modularized among agents; one agent may have to re-plan, but the others can remain with their previous plans.

As in the previous scenario, at each step, all agents state additional information about the sub-plan of which they are in charge. The next l optimal steps are then determined and the current configuration of the world, s^t, is changed to be s^{t+1}. The process continues until all tasks have been accomplished (the global goal as of that specific time has been achieved).

Since steps of the plan are executed in parallel to the planning process, the smaller the look-ahead factor is, the smaller the weight of the g component of the evaluation function becomes (and the more the employed search method resembles Hill Climbing). Therefore, the resulting multi-agent plan may only approximate the actual optimal global plan.

Theorem 2 *Let the cost effect of "positive" interactions among members of some subset, p, of the set of sub-plans, P^t, that achieves G^t be denoted by δ_p^+, and let the cost effect of "negative" interactions among these sub-plans be denoted by δ_p^-. Accordingly, let $\delta = \max_{p \in P^t} \mid \delta_p^+ - \delta_p^- \mid$.*[7] *We say that the multi-agent plan that achieves G^t is δ-optimal, if it diverges from the optimal plan by at most δ.*

Then, at any time step t, employing the merging algorithm, the agents will follow a δ-optimal multi-agent plan that achieves G^t.

[7]The effect of heuristic overestimate (due to positive future interaction between individual plans) and the effect of heuristic underestimate (due to interference between individual plans) offset one another.

Conclusions and Related Work

In this paper, we presented a heuristic multi-agent planning framework. The procedure relies on an *a priori* division of the global goal into subgoals. Agents solve local subgoals, and then merge them into a global plan. By making use of the computational power of multiple agents working in parallel, the process is able to reduce the total elapsed time for planning as compared to a central planner. The optimality of the procedure is dependent on several heuristic aspects, but in general increased effort on the part of the planners can result in superior global plans.

An approach similar to our own is taken in (Nau, Yang, & Hendler 1990) to find an optimal plan. It is shown there how planning for multiple goals can be done by first generating several plans for each subgoal and then merging these plans. The basic idea there is to try and make a global plan by repeatedly merging complete plans that achieve the separate subgoals and answer several restrictions. In our approach, there are no prior restrictions, the global plan is created incrementally, and agents do the merging in parallel.

In (Foulser, Li, & Yang 1992) it is shown how to handle positive interactions efficiently among different parts of a given plan. The merging process looks for redundant operators (as opposed to aggregating propositions) within the same *grounded linear plan* in a dynamic fashion. In (Yang 1992), on the other hand, it is shown how to handle conflicts efficiently among different parts of a given plan. Conflicts are resolved by transforming the planning search space into a constraint satisfaction problem. The transformation and resolution of conflicts is done using a backtracking algorithm that takes cubic time. In our framework, both positive and negative interactions are addressed simultaneously.

Our approach also resembles the GEMPLAN planning system (Lansky & Fogelsong 1987; Lansky 1990). There, the search space is divided into "regions" of activity. Planning in each region is done separately, but an important part of the planning process within a region is the updating of its overlapping regions (while the planning process freezes).

Our planning framework also relates to the approach suggested in (Wellman 1987). There too the planning process is viewed as the process of incremental constraint posting. A method is suggested for assigning preferences to sets of constraints (propositions in our terminology) that will direct the planner. However, the evaluation and comparison between alternatives is done according to the global view of the single planner, and is based on pre-defined dominance relations.

Acknowledgments

This work has been supported in part by the Air Force Office of Scientific Research (Contract F49620-92-J-0422), by the Rome Laboratory (RL) of the Air Force Material Command and the Defense Advanced Research Projects Agency (Contract F30602-93-C-0038), and by an NSF Young Investigator's Award (IRI-9258392) to Prof. Martha Pollack.

And in part by the Israeli Ministry of Science and Technology (Grant 032-8284).

References

Dean, T., and Boddy, M. 1988. An analysis of time-dependent planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, 49–54.

Durfee, E. H. 1990. A cooperative approach to planning for real-time control. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, 277–283.

Ephrati, E. 1993. *Planning and Consensus among Autonomous Agents.* Ph.D. Dissertation, The Hebrew University of Jerusalem, Jerusalem, Israel.

Foulser, D. E.; Li, M.; and Yang, Q. 1992. Theory and algorithms for plan merging. *Artificial Intelligence* 57:143–181.

Korf, R. E. 1987. Planning as search: A quantitative approach. *Artificial Intelligence* 33:65–88.

Lansky, A. L., and Fogelsong, D. S. 1987. Localized representation and planning methods for parallel domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 240–245.

Lansky, A. L. 1990. Localized search for controlling automated reasoning. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, 115–125.

Martial, F. von 1990. Coordination of plans in multiagent worlds by taking advantage of the favor relation. In *Proceedings of the Tenth International Workshop on Distributed Artificial Intelligence.*

McAllester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 634–639.

Minton, S.; Bresina, J.; and Drummond, M. 1991. Commitment strategies in planning: A comparative analysis. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, 259–265.

Nau, D. S.; Yang, Q.; and Hendler, J. 1990. Optimization of multiple-goal plans with limited interaction. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, 160–165.

Penberthy, J., and Weld, D. 1992. UCPOP: A sound, complete, partial order planner for ADL. In *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, 103–114.

Wellman, M. P. 1987. Dominance and subsumption in constraint-posting planning. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, 884–889.

Yang, Q. 1992. A theory of conflict resolution in planning. *Artificial Intelligence* 58(1-3):361–393.