

doi: 10.7690/bgzdh.2016.11.021

基于复杂网络的作战任务分发流程优化研究

李俊^{1,2}, 严建钢³, 杨士锋³, 何颀¹

(1. 海军航空工程学院研究生管理大队, 山东 烟台 264001;

2. 中国人民解放军 92913 部队, 海南 临高 571820;

3. 海军航空工程学院指挥系, 山东 烟台 264001)

摘要: 为提高作战指挥系统组织部队进行战斗准备的速度, 对基于复杂网络的作战任务分发流程优化进行研究。建立整个作战系统模拟网络模型, 给出每个节点时间的图解算法, 根据各个节点时间值对于整个系统反应时间的影响的大小, 得出系统关键线路节点与关键线路, 最后通过调整关键节点任务分发流程, 得到较快的作战指挥系统反应时间, 并进行算例仿真。仿真结果表明: 该方法能够准确解算出作战指挥系统的反应时间, 得出局部最优解。

关键词: 作战任务; 网络; 流程**中图分类号:** TP391.9 **文献标志码:** A

Optimization Research of Combat Mission Distribution Procedure Based on Complex Net

Li Jun^{1,2}, Yan Jian'gang³, Yang Shifeng³, He Xie¹(1. *Administrant Brigade of Postgraduate, Naval Aeronautical Engineering Academy, Yantai 264001, China;*2. *No. 92913 Unit of PLA, Lingao 571820, China;*3. *Department of Command, Naval Aeronautical Engineering Academy, Yantai 264001, China)*

Abstract: In order to enhance the speed of organizing army to get ready for battle by combat command system, research on the optimization of combat mission's distribution procedure based on complex net. Build a model for whole combat system simulation net, give diagram algorithm of every node time, according to influence of node time value on system response time, acquire system key line node and key line. At last, by adjusting key node mission distribution procedure, acquire fast combat command system response time and carry out example simulation. The simulation results show that the method can accurately calculate response time of combat command system and acquire local optimal solution.

Keywords: combat mission; net model; procedure

0 引言

目前针对系统的流程研究大多是基于缩短武器系统和部队作战反应时间的流程优化。文献[1]研究怎样通过优化流程缩短部队反应时间的统筹方法; 文献[2-3]分析研究空降兵防空分队与海上作战分队作战反应时间, 需要包含的任务流程, 虽没有具体模型, 但提出了优化流程对缩短作战反应时间有较大意义; 文献[4]对武器系统的反应时间进行了研究; 文献[5-6]构建了作战体系网络模型, 分析了作战体系网络的基本特征; 文献[7]给出了基于固定指挥网络下作战指挥系统反应时间的概率值分析, 但没有研究调整网络结构对作战反应时间的影响。

以复杂网络模型为基础, 设定命令发出点为网络的起点, 整个作战指挥系统执行这一命令, 命令的完成终端作为网络的终点, 则作战指挥系统可看作是一个从上之下的有向网络, 不同指挥网络结构的指挥系统完成任务的时间也不同, 如何通过调整

任务分发流程使完成任务时间最短, 目前较流行的是采用图论解法^[8]与关键路线法^[9]来解决, 笔者建立指挥网络的邻接矩阵, 然后利用 Matlab 软件进行仿真计算, 最后利用调整关键线路的方法对作战任务的分发流程进行优化, 得到较快的系统反应时间。

1 模型的建立

1.1 作战指挥系统网络

如图 1 所示, 节点 1 为命令的起点, 可以模拟一个单位系统的任务接收节点, N 为终点, 即执行完毕所受领任务。

阶段定义: 整个网络系统为从 1 节点到 N 节点的定向运动, 图 1 中任意节点都存在向其运动的节点(定义为该节点的上阶段节点)与该节点需要运动至其他节点(定义为该节点的下阶段节点)。可知, 网络中节点 1 没有上阶段节点, 节点 N 没有下阶段节点, 其余节点均存在上阶段节点与下阶段节点。

收稿日期: 2016-07-07; 修回日期: 2016-08-27

作者简介: 李俊(1985—), 男, 四川人, 硕士, 从事作战模型与模拟研究。

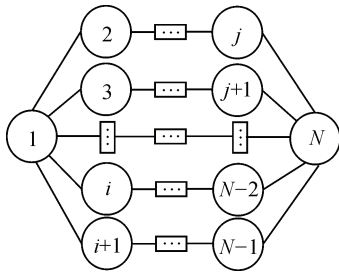


图 1 作战指挥系统网络

任务分发流程是指在作战指挥中，一个单位在一定时间内需要完成多项任务，而完成任务的先后顺序即为分发流程。而对于图 1 的网络模型，即模型中一个节点存在多个下阶段节点的情况，而其完成下阶段节点的先后顺序即为任务分发流程。

1.2 模型时间图解法

如图 1 所示，依照从左至右，从上到下依次编号 1,2,3,...,i,...,j,...,N，即整个网络有 N 个节点，设第 i 个节点任务的完成所需要的时间为 T_i ，其到下一个节点 j 所需要的时间为 t_{ij} ，其意义为从 i 节点完成到 j 节点完成第 i 个节点所做工作需要花费的时间，作战指挥系统反应时间如图 2 所示。

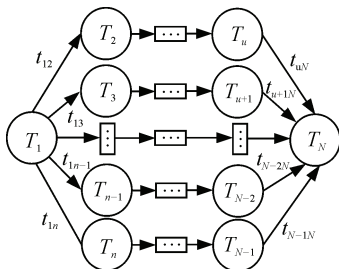


图 2 作战指挥系统反应时间

在此假设整个系统中没有如图 3 所示的闭环。

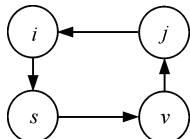


图 3 闭环示意

如果已知 T_i 与 t_{ij} ， T_j 的计算有 4 种情况：

1) “一对一”情况，即 i 的下一阶段节点只有一个，而 j 的上一节点也只有一个，如图 4 所示。

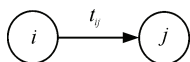


图 4 “一对一”示意图

在此情况下，j 节点的时间值按照下式计算：

$$T_j = T_i + t_{ij} \quad (1)$$

2) “一对多”情况，即 i 的下一阶段节点有多个，而 j 的上一节点只有一个，如图 5 所示。

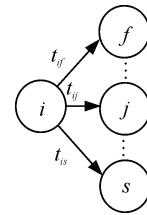


图 5 “一对多”示意图

假设 i 所对应的下一阶段有 $(f, f_1, \dots, j, \dots, s)$ 若干个节点，在这种情况下， T_j 的值取决于它在这些节点中执行任务的顺序位置，即任务分发流程，如果节点 i 按照从上至下依次进行任务分发，那么

$$\left. \begin{aligned} T_f &= T_i + t_{if} \\ T_{f-1} &= T_f + t_{if_1} \\ &\vdots \\ T_j &= T_{j-1} + t_{ij} \\ &\vdots \\ T_s &= T_{s-1} + t_{is} \end{aligned} \right\} \quad (2)$$

把式(2)中从上至下依次带入下一个式子的右边，可得

$$T_j = T_i + \sum_{k=f}^j t_{ik} \quad (3)$$

令 $t'_{ij} = \sum_{k=f}^j t_{ik}$ ，式(3)变成

$$T_j = T_i + t'_{ij} \quad (4)$$

可以看出式(4)与式(1)相似。

3) “多对一”的情况，即 j 的上一阶段节点有多个，而 i 的下一节点只有一个，如图 6 所示。

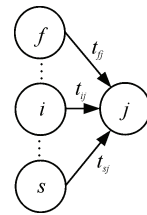


图 6 “多对一”示意图

假设 j 的上一阶段有 f 到 s ($s > f$) 个节点，那么只有当 T_f 至 T_s 全部完成工作后，才能算着 j 完成，如果 f 至 s 完成先后顺序对 T_j 的完成不产生影响，那么

$$T_j = \max_{f \leq i \leq s} (T_i + t_{ij}) \quad (5)$$

在实际工作中，j 的完成经常需要 f 至 s 按照一定的先后顺序完成。例如，事件 f 的完成需要早于事件 s 的完成，提前时间为 Δt_{fs} ，事件 s 才能向事件 j 转移，同时事件 f 与事件 s 并没有阶段上的前后关系，2 个事件都可以独立完成，但在向 j 转移时

需要一定的先后顺序。在计算 T_j 时应该注意以下 2 种情况:

一是当 $T_s - T_f < \Delta t_{fs}$ 时, 将 T_s 重新赋值

$$T_s = T_f + \Delta t_{fs} \quad (6)$$

二是当 $T_s - T_f \geq \Delta t_{fs}$, T_s 值保持不变。按照这种原则整理后, 再按式(5)计算 T_j 值。

4) “多对多”的情况, 即 j 的上一阶段节点有多个, 而 i 的下一阶段节点有多个, j 只是 i 的下阶段节点中的一个, 而 i 只是 j 众多上阶段节点中的一个。假设 j 的上一阶段节点为 f 至 s , T_i 的下一阶段节点为 f_1 至 s_1 , 如图 7 所示。

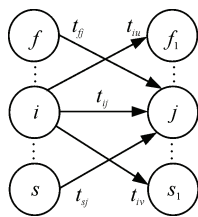


图 7 “多对多”示意图

这种情况可以分解为“一对多”与“多对一”2 种情况的组合, 可以先把此种关系转化为“一对多”情况, 根据任务分发流程先计算出 t'_{ij} 的值代替 t_{ij} 的值, 然后根据“多对一”情况求 T_j 。

从上述分析来看, 整个作战指挥网络节点关系无外乎以上 4 种情况。如果假设接受信息原点的时间定为 0, 各个节点到其下一个节点的所需要花费的时间都为已知, 且针对“多对一”情况, 一个节点存在多个上一阶段节点, 节点的先后顺序也为已知, 那么就可以根据上面模型从前至后求出每个节点的完成时间值, 最后求得 T_N 的值, T_N 即为整个作战指挥系统的反应时间。

2 模型的求解

模型的求解思路是根据已知网络建立起各节点间的邻接矩阵, 利用邻接矩阵来实现上述网络算法。

2.1 建立模型邻接矩阵

由图 2 可以看出: 将整个系统看着有着 N 个节点网络, 而且是一个有向图, 是从第 1 个节点流向第 N 个节点, 图形可以由矩阵 $A^{[10]}$ 表示

$$A = \begin{pmatrix} 0 & t_{12} & \cdots & t_{1N-1} & t_{1N} \\ t_{21} & 0 & \cdots & t_{2N-1} & t_{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{N-11} & t_{N-12} & \cdots & 0 & t_{N-1N} \\ t_{N1} & t_{N2} & \cdots & t_{NN-1} & 0 \end{pmatrix} \circ$$

上面矩阵为图 2 的邻接矩阵, 矩阵的元素 $A(i, j)$ 表示第 i 个节点到第 j 个节点需要花费的时间, 令 $A(i, j) = t_{ij}$ 。规定节点自身不花费时间, 而由于整个图为一个有向图, 如果 i 节点与 j 节点存在一条边相连, 且 i 节点为 j 节点上一级节点, 那么 $A(i, j)$ 有意义, 而 $A(j, i)$ 没有意义, 即 $A(i, j)$ 与 $A(j, i)$ 其中之一有意义; 如果 2 个节点不存在边相连, 那么 $A(i, j)$ 与 $A(j, i)$ 都没有意义, 没有意义的元素用 inf 表示, 在计算中将不为 0 和 inf 的元素定义为有效元素 ($A(i, j) \neq 0, A(i, j) \neq \text{inf}$)。易知, 矩阵 A 为稀疏矩阵。且有如下 4 种情况:

- 1) 当 i 行只有一个有效元素 $A(j, i)$, 第 j 列也只有一个有效元素 $A(j, i)$ 时, 此时为“一对一”情况;
- 2) 当 i 行有多个元素, 且第 j 列只有一个元素时, 此时为“一对多”情况;
- 3) 当第 j 列有多个元素, 且有效元素对应行只有一个元素时, 此时为“多对一”情况;
- 4) 当 i 行只有多个元素, 且第 j 列只有多个元素时, 此时为“多对多”情况。

2.2 建立任务提前矩阵

建立一个与 A 同阶矩阵 Δ , 将 A 中元素为无效元素的相同位置, 在 Δ 中设为无效元素 inf , A 中有效元素相同位置设为 0。

依次查找矩阵 A 的第 $j(j=1, 2, \dots, N)$ 列有效元素的个数 σ , 当 $\sigma > 1$ 时, 表示第 j 个节点的上一阶段节点有 σ 个, 按照实际要求, 如果上一阶段节点完成存在先后顺序时, 确定一个需要最先完成的节点, 设为节点 p 为必须最先完成, 将 $\Delta(p, j)$ 设为 0, 其余按 p 对其的提前时间量来设置, 例如事件 p 必须先于事件 k 完成的时间为 Δt_{pk} , 那么 $\Delta(k, j) = \Delta t_{pk}$, 如果没有完成时间的上先后顺序, 则设为 0。

2.3 图解法的矩阵实现

首先建立一维数组 $T_{1 \times N}$, $T(i)(i \in 1, \dots, N)$ 为第 i 个节点的时间值, 将 $T(1)$ 设为 0, $T(N)$ 即为整个指挥系统的反应时间值。与图解法对应, $T(i)$ 的计算分为下面 4 种情况

2.3.1 “一对一”情况的矩阵实现

当矩阵 A 的第 $i(i \in 1, 2, \dots, N)$ 行有效元素个数为 1, 设有效元素为 $A(i, j)$, 且 A 的第 j 列有效元素的

个数也为 1，那么，按照“一对一”图解法可知

$$T(j) = T(i) + A(i, j) \quad (7)$$

式中 $T(i)$ 、 $T(j)$ 为一维数组 $T_{1 \times N}$ 的第 i 、 j 个元素值，后同。

2.3.2 “一对多”情况的矩阵实现

当矩阵 A 的第 $i(i \in 1, 2, \dots, N)$ 行有效元素个数为 $u(u > 1)$ ，分别为 $\underbrace{A(i, f), \dots, A(i, s)}_{u \uparrow}$ ，按照“一对多”情况求解，先确定一个任务分发流程，即由 i 节点确定一个要执行任务的先后顺序，假如先后顺序为 $\psi^i = (f, \dots, j, \dots, s)$ 从前至后依次进行，那么

$$T(j) = T(i) + \sum_{k=f}^j A(i, k) \quad (8)$$

并记录下任务分发流程数组 ψ^i ，整个系统存在若干“一对多”的情况，将这些分发流程组合起来形成整个作战指挥系统的分发流程数组 $\psi = \{\psi^i\} (i \in (1, \dots, N))$ 。

2.3.3 “多对一”情况的矩阵实现

当矩阵 A 的第 $j(j \in 1, 2, \dots, N)$ 列有效元素个数为 $v(v > 1)$ ，分别为 $\underbrace{A(f', j), \dots, A(s', j)}_{v \uparrow}$ ，且矩阵 A 的第 f', \dots, s' 行只有唯一有效元素，按照“多对一”情况求解，在此已知 j 节点所有上阶段节点的时间值 $T(f'), \dots, T(s')$ ，设 i 与 p 均为节点 j 的上一阶段节点，在所有上阶段节点中， p 为必须先行完成节点，按

$$T'(i) = \max(T(i), T(p) + \Delta(i, j)) \quad (9)$$

重新对 $T(f'), \dots, T(s')$ 的值进行修正，记为 $T'(f'), \dots, T'(s')$ ，式(10)中 $\Delta(i, j)$ 为任务提前矩阵 Δ 中元素 $\Delta(i, j)$ 的值。修正后， $T(j)$ 按照下式进行计算。

$$T(j) = \max(T'(k) + A(k, j), k \in f', \dots, i, \dots, s') \quad (10)$$

2.3.4 “多对多”情况的矩阵实现

当矩阵 A 的第 $j(j \in 1, 2, \dots, N)$ 列有效元素个数为 $v(v > 1)$ ，且对应有效元素所在行的有效元素不唯一或部分不唯一，要求 $T(j)$ 的值，此时为“多对多”情况求解，首先按照 2.3.2 节“一对多”情况进行处理，假设确定若干个任务分发流程为 $\psi^i = (f, \dots, j, \dots, s) i \in (f', \dots, i, \dots, s')$ ， i 为任意存在“一对多”形式 j 的上阶段节点。若顺序为从前至后依次执行，令

$$T^i(j) = T(i) + \sum_{k=f}^{j-1} A(i, k) \quad (11)$$

$T^i(j)$ 的数量与节点 j 的上阶段节点的任务分发流程对的数量一致，例如 j 节点同时存在于 2 个不同的分发流程当中，分别为 i 节点与 f' 节点的“一对多”节点，那么就有同时存在 $T^i(j)$ 与 $T^{f'}(j)$ 。再按照“多对一”的方法，将式(10)变为

$$T'(i) = \max(T^i(j), T^{f'}(j) + \Delta(i, j)) \quad (12)$$

重新对 $T(f'), \dots, T(s')$ 的值进行修正，记为 $T'(f'), \dots, T'(s')$ ， $T(j)$ 值按照式(11)进行计算。

最后得到数组 $T_{1 \times N}$ 的最后一个元素 $T(N)$ ，为整个系统在作战任务分发流程为 ψ 时的反应时间。

2.4 模型的优化算法

2.4.1 关键线路

通过 2.3 节的计算，可以得到整个作战指挥系统的反应时间 $T(N)$ 以及相对应的作战任务分发流程 ψ 。根据文献[1]可知：关键线路作为整个网络系统中主要矛盾线路，其耗时最长，调整关键线路算法主要思想就是适当调整关键节点在任务分发流程中的位置，减少关键线路用时，以此减少整个系统的反应时间，关键线路如图 8 中加粗实线所示。

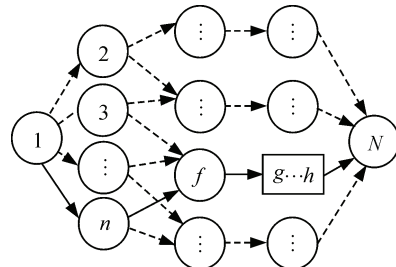


图 8 关键线路

关键线路是由关键节点和连接关键节点的边连接而成的从起点到终点的一条线路，寻找关键线路的基础是找到关键节点。在以上算法中，笔者采用逆推法寻找关键节点。在 2.3 节中得到一个一维数组 $T_{1 \times N}$ ，首先确定节点 N 为关键节点，其节点时间值 $T(N)$ ，首先令 $\Omega = N$ ，按照以下步骤求关键节点：

Step1: 确定关键节点 Ω ，与关键节点的时间值 $T(\Omega)$ ，在矩阵 A 中找到相应的第 Ω 列，查找第 Ω 列的所有有效元素，并记录每个元素的行标 $(\alpha, \dots, \phi, \dots, \beta)$ ；

Step2: 如果 $T(\Omega) = T(\phi) + t_{\phi\Omega}$ ，且 $T(\phi)$ 没有经过式(10)与式(13)修正，即 $T'(\phi) = T(\phi)$ 或 $T'(\phi) = T^\phi(\Omega)$ ，

那么 ϕ 就为 Ω 节点的上一阶段的关键节点, 令 $\Omega = \phi$, 记录下 ϕ , 返回 step1;

Step3: 如果 $T'(\Omega) = T(\phi) + t_{\phi\Omega}$, 且 $T(\phi)$ 经过式(10)或式(13)修正, 即 $T'(\phi) \neq T(\phi)$ 或 $T'(\phi) \neq T^\phi(\Omega)$, 那么必须提前完成的 p 节点就为 Ω 节点的上一阶段的关键节点, 令 $\Omega = p$, 记录下 p , 返回 step1;

Step4: 当 $\Omega = 1$ 时, 跳出。

根据以上算法, 可得到一组记录的节点即为系统的关键节点。从算法上可看出: 关键节点的时间值与其上阶段的关键节点时间值和两阶段关键节点之间连线的权值(节点转移需要的时间值)有关系, 所以两阶段关键节点之间的连线视为 2 个阶段间的关键线路, 从节点 N 从后往前逐个连接关键节点的一条贯穿整个系统的线路即为系统的关键线路。

2.4.2 调整关键线路

根据式(9)和式(12)可知, 适当调整关键线路上任务的分发流程, 可以得到更小关键节点的时间值, 从而降低整个系统的反应时间, 优化算法如下:

Step1: 按照上面的算法, 先算出整个系统的反应时间 $T(N)$, 以及关键线路 $\gamma_{\text{key}} = (1, n, f, \dots, N)$, 令 $\xi = T(N)$, 并确定数组集合 ψ ; 设立 2 个参数 $\vartheta = 2, \zeta = 1$;

Step2: 将关键线路进行逆序排列, $\gamma'_{\text{key}} = (N, s, \dots, f, n, 1)$, 在在数组集合 ψ 中找出含有关键节点 $\gamma'(\vartheta)$ (表示 γ' 中第 ϑ 个元素)的数组, 设为

$$[(i, \dots, \gamma'(\vartheta), \dots, j), \dots, (i', \dots, \gamma'(\vartheta), \dots, j')]$$

d 组

将这些数组重新定义为 ψ' ;

Step3: 在 ψ' 的第 ζ 个数组中, 将关键节点的位置与其前一位交换, 后用 ψ' 替换 ψ 中对应的数组;

Step4: 根据新的任务分发流程数组 ψ , 按照 2.3 节的方法求出新的 $T(N)$, 如果 $T(N) < \xi$, 令 ξ 等

于新的 $T(N)$, 转 Step3; 如果 $T(N) > \xi$, 在 ψ' 的第 ζ 个数组中, 将关键节点与其后一个节点交换, 然后用 ψ' 替换 ψ 中对应的数组, $\zeta = \zeta + 1$, 转 Step3; 如果 $\zeta = d$, 转 Step5;

Step5: 如果 $\gamma'(\vartheta) = 1$, 跳出, 记录下 ξ , 如果 $\gamma'(\vartheta) \neq 1, \vartheta = \vartheta + 1$, 转 Step2。

在进行完这 5 个步骤至后, 可以得到仍然以 $\gamma_{\text{key}} = (1, n, f, \dots, N)$ 为关键节点的系统最短时间 (ξ) 与最优任务分发流程 ψ , 在随机选取 k 种分发流程, 按照以上算法可得到 $\kappa (\kappa \leq k)$ 个关键线路、 κ 个系统最短反应时间和 κ 组最优任务分发流程, 在 κ 个系统最短反应时间内选取最小值作为算法寻找的系统反应较小值, 与其对应的任务分发流程即为作战系统的较优作战任务分发流程, 即局部最优解。

3 算例仿真

为了验证算法的正确性, 选一个 27 个节点的系统进行验证。如图 9, 节点 1 为作战指挥系统命令的接收点, 节点 27 为作战指挥系统完成这一命令。

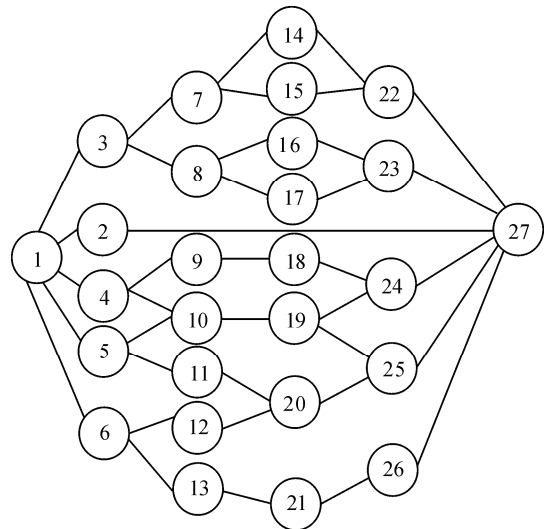


图 9 作战指挥系统网络

表 1 系统的邻接矩阵

坐标	元素	坐标	元素	坐标	元素	坐标	元素
(1,2)	0.500 0	(4,10)	4.308 0	(10,19)	0.405 2	(19,25)	0.106 9
(1,3)	0.500 0	(5,10)	5.543 6	(11,20)	2.470 1	(20,25)	5.227 5
(1,4)	0.500 0	(5,11)	6.452 7	(12,20)	5.692 2	(21,26)	3.115 7
(1,5)	0.500 0	(6,12)	5.167 5	(13,21)	0.069 0	(22,27)	6.522 7
(1,6)	0.500 0	(6,13)	1.233 9	(14,22)	0.972 2	(23,27)	3.262 0
(2,27)	5.334 7	(7,14)	2.839 9	(15,22)	1.419 4	(24,27)	2.930 5
(3,7)	3.195 3	(7,15)	6.548 3	(16,23)	1.391 1	(25,27)	5.923 5
(3,8)	0.129 5	(8,16)	6.418 3	(17,23)	4.226 5	(26,27)	3.676 1
(4,8)	5.749 9	(8,17)	2.871 9	(18,24)	1.905 3		
(4,9)	3.112 9	(9,18)	6.255 5	(19,24)	1.391 7		

根据第 2 节介绍, 图 9 所示系统可表示为 27×27

的临接矩阵, 矩阵为主对角线都为 0 的稀疏矩阵,

为了不失一般性,矩阵有效元素采用 0~7 之间的随机数来代替,时间为单位时间。矩阵如表 1 所示。

矩阵的第一行有效元素为 0.5,是指作战值班室接到命令后,通知 5 个部门(部门 2, 3, 4, 5, 6),通知时间 0.5 个单位时间。

为了表示在多个节点共同完成一项任务,且不同节点完成的先后顺序不同,按照 2.3 节的方法随机建立任务提前矩阵,如表 2 所示,任务提前矩阵的元素同样为随机取得。

表 2 各项任务的提前矩阵

坐标	元素	坐标	元素	坐标	元素	坐标	元素
(2,27)	2	(11,20)	1	(17,23)	1	(22,27)	0
(3,8)	3	(12,20)	0	(18,24)	0	(23,27)	2
(4,8)	0	(14,22)	0	(19,24)	3	(24,27)	1
(4,10)	2	(15,22)	2	(19,25)	2	(25,27)	2
(5,10)	0	(16,23)	0	(20,25)	0	(26,27)	1

该矩阵除了以上元素之外,其余都是无效元素,第 10 列的 2 个有效元素 2、0,表示 4、5 节点共同完成 10 节点代表的任务,但节点 5 要比节点 4 先完成 2 个单位。表 1 与表 2 都可以根据具体的部队与任务在现实中确定,由于保密需求与模型的通用性要求,这里采取的是随机选取的数据进行算法仿真。从图 9 可以得出整个系统共有 8 个作战任务分发流程对,分别是:

1 → (2,3,4,5,6); 3 → (7,8); 4 → (8,9,10); 5 → (10,11);
6 → (12,13); 7 → (14,15); 8 → (16,17); 19 → (24,25)。

笔者分别采取枚举法、调整关键线路算法、遗传算法,用 Matlab 软件对上述系统进行仿真计算。

枚举法的计算次数为 $\prod_{i=1}^d A_i^s$ 次, d 代表整个系统有 d 个作战任务分发流程对,表示其中一个分发流程对中有 s 个节点, A_i^s 表示一个全排列。如果采取枚举法,本算例的计算次数为 46 080 次;所以随机抽取 81 种任务分发流程进行计算,结果如表 3 所示。

表 3 几种算法对系统反应时间值计算的比较

计算方法	系统反应时间最小值
枚举法(随机 81 个)	25.028 6
调整关键线路算法	24.244 6
遗传算法	23.606 4

按照调整关键线路算法得到的较优作战任务分发流程为:

$$\left[(4,6,5,3,2); (7,8); (8,9,10); (11,10); (13,12); (14,15); (16,17); (25,24) \right]。$$

作战指挥系统最短反应时间为 24.244 6 个单位。采用遗传算法得到的较优作战任务分发流程为:

$$\left[(4,6,3,5,2); (8,7); (8,9,10); (11,10); (12,13); (14,15); (16,17); (25,24) \right]，$$

系统反应的最小值为 23.606 4 个单位。

从计算结果可以看出:调整关键线路法比一般的枚举法可以得到较快的系统反应时间,虽然相对于遗传算法还是有微小差距,但根据笔者实验,在使用同一台计算机进行仿真计算时,采用调整关键线路算法可以在 1 s 左右得到该方法的较优解,而用遗传算法的计算时间为 17 min 以上,故在一些需要快速计算且对解要求不高的情况下,调整关键线路算法有一定的优越性。

4 总结及展望

笔者建立了作战系统的网络模型,并利用等效的邻接矩阵解算出作战系统的反应时间,利用调整关键线路算法得出局部最优解,但只考虑在每个节点完成一定任务需要的时间为定值,在现实情况下,可通过加大人力物力投入的等方式去降低任务完成时间,这样就可进一步加快系统的反应时间。但对于不同任务投入与减少的时间关系并不一样,所以将人力物力投入加到哪些节点还需要进一步优化,同时,可以进一步研究如何改进遗传算法,降低计算时间,使其能快速地接近全局最优解。

参考文献:

- [1] 谭乐祖, 翟军. 军事运筹学教程[M]. 北京: 兵器工业出版社, 2010: 239-272.
- [2] 齐和平. 空降兵防空分队作战反应时间[J]. 火力指挥与控制, 2013, 38(9): 175-176.
- [3] 李孟启. 对缩短海上作战指挥反应时间的思考[J]. 海军学术研究, 2004(3): 29-30.
- [4] 杨兴宝. 基于随机 Petri 网的舰空导弹武器系统作战反应时间研究[J]. 战术导弹技术, 2012(6): 38-41.
- [5] 王庆功, 沈寿林, 袁昌平. 基于复杂网络的作战体系网络模型分析[J]. 指挥控制与仿真, 2013, 35(1): 6-10.
- [6] 张宪. 基于复杂网络的联合作战指挥网络拓扑结构[J]. 指挥控制与仿真, 2013, 36(5): 1-5.
- [7] 宋元, 章新华. 作战指挥网络的作战反应时间分析[J]. 海军大连舰艇学院学报, 2003, 26(3): 16-18.
- [8] 韦新, 邓天炎, 李碧荣. 图论模型及其应用[J]. 兰州文理学院学报(自然科学版), 2014, 28(4): 38-42.
- [9] 伊鑫, 张伟, 王忠思, 等. 统筹法在军事通信系统开发中的应用[J]. 兵工自动化, 2012, 31(5): 94-96.
- [10] 管志忠, 刘永明. 图论中最短路问题的 Matlab 程序实现[J]. 安庆师范学院学报(自然科学版), 2007, 13(1): 26-29.